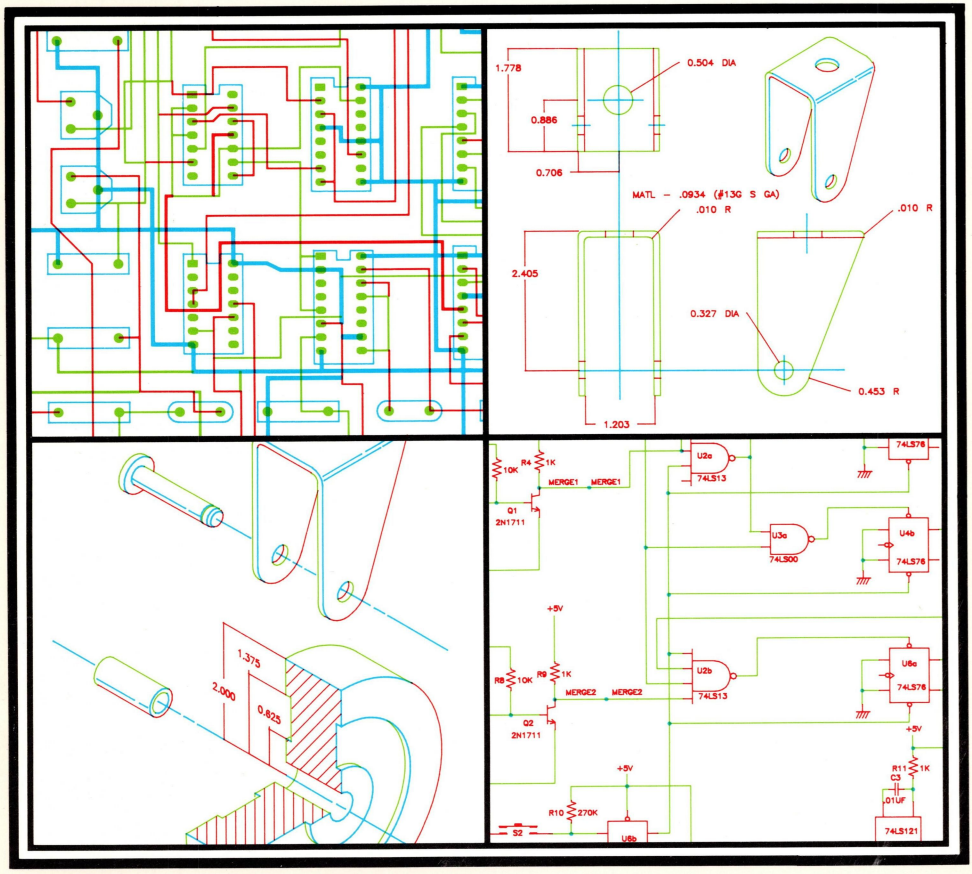
HP 9000 Series 200/300 Computers

HEWLETT PACKARD

HP EGS 2.2 Syntax Reference



HP EGS Manuals

Installing HP EGS lists the hardware and software needed for an HP EGS system, gives some hints on connecting the hardware, and explains installing the HP EGS software on a hard disc and Shared Resource Manager. Appendices contain information on using a hard disc and Shared Resource Manager, the differences between HP EGS 2.2 and 2.1, combining HP EGS 2.2 with a complete Pascal 3.12 Operating System, and using a LaserJet printer with HP EGS 2.2.

Learning HP EGS provides tutorials to familiarize you with the HP EGS system and its modules: Engineering Graphics, Electrical Schematic Drawing, Printed Circuit Board Layout, and Mechanical Drafting. By following the exercises in this manual you will learn the basic concepts of HP EGS so you can do more advanced tasks easily.

Understanding HP EGS describes the theory and operation of the HP EGS system. It explains how the Graphics Editor works and how to customize the system. You can learn how to use HP EGS post-processors to create and manipulate material and connection lists, produce output to run photoplotter and drill machines, and translate drawings to the EDIF Translator and to and from the IGES Translator. Application notes on using archive files to transport data, using the IGES Translator, and writing macros are also included.

Managing and Editing Files with HP EGS describes file system concepts and provides tutorial exercises to help you learn how to use the text Editor and Filer. A syntax reference further explains the Editor, Filer, and Pascal Command Line commands.

HP EGS Syntax Reference lists all HP EGS commands. Each command is illustrated with a diagram, explained in a table, and shown in use with one or more examples. The introductory sections of the reference explain how to interpret the diagrams and enter the commands. The commands used in archive files are listed at the end of the reference.

HP EGS 2.2 Syntax Reference

HP 9000 Series 200/300 Computers

HP Part Number 98305-90008

Hewlett-Packard Company
3404 East Harmony Road, Fort Collins, Colorado 80525

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MANUAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

WARRANTY

A copy of the specific warranty terms applicable to your Hewlett-Packard product and replacement parts can be obtained from your local Sales and Service Office. For HP EGS 98305 Option 085 warranty information see the last page of the *Installing HP EGS* manual.

Copyright 1985, 1986 Hewlett-Packard Company

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced or translated to another language without the prior written consent of Hewlett-Packard Company. The information contained in this document is subject to change without notice.

Restricted Rights Legend

Use, duplication or disclosure by the Government is subject to restrictions as set forth in paragraph (b)(3)(B) of the Rights in Technical Data and Software clause in DAR 7-104.9(a).

Use of this manual and flexible disc(s) or tape cartridge(s) supplied for this pack is restricted to this product only. Additional copies of the programs can be made for security and back-up purposes only. Resale of the programs in their present form or with alterations, is expressly prohibited.

Printing History

New editions of this manual will incorporate all material updated since the previous edition. Update packages may be issued between editions and contain replacement and additional pages to be merged into the manual by the user. Each updated page will be indicated by a revision date at the bottom of the page. A vertical bar in the margin indicates the changes on each page. Note that pages which are rearranged due to changes on a previous page are not considered revised.

The manual printing date and part number indicate its current edition. The printing date changes when a new edition is printed. (Minor corrections and updates which are incorporated at reprint do not cause the date to change.) The manual part number changes when extensive technical changes are incorporated.

November 1986...Edition 1

Table of Contents

Using this Reference

Interpreting the Syntax Diagrams 1
Using Component Abbreviations 2
Understanding the Types of Commands 3
Entering Commands 4
    Selecting Snapping Modes 4
    Completing Primary Commands 5
    Using the HP EGS Module Commands 6
ADD 7
ADD ARC 10
ADD CIRCLE 14
ADD DIMENSION 18
ADD HATCH 24
ADD instance 30
ADD LINE 32
ADD MARKER 39
ADD OVAL 40
ADD POLYGON 44
ADD RECTANGLE 46
ADD TEXT or NOTE 48
ARCHIVE 53
AREA 56
BACK_ANNOTATE 58
^ (backup) 62
CALIBRATE 63
CATALOG 66
CHANGE 68
COPY 70
CURSOR 76
DATUM 81
DECIMALS 82
DEFINE 84
DELETE 90
DIM_DEC 92

DIM_DIR 93
DIM_ENTER 94
DIM_FSIZE 96
DIM_RADIX 97
DIM_SCALE 99
DIM_TFSIZE 100
DIM_TSLANT 101
DIM_UNITS 102
DISPLAY 104
DISTANCE 106
DUMP 107
ECHO 109
EDIT 110
EQUATE 116
EVALUATE 120
EXIT 123
FILECOPY 124
FSIZE 126
GATHER 127
GENERATE 134
GRID 138
GROUP 142
HELP 145
IDENTIFY 149
INAME 151
INPUT 152
LEVEL 155
LIST 156
LOAD 159
LOCATOR 163
LOCK 164
MEMORY 167
MENU 168
MODIFY 172
MOVE 176
NOP 178
PACK 179
PAGE 180
PAUSE 181
PLOT 182
POINT 191

PREFIX 192
QUIT 193
REMOVE 194
RESOLUTION 196
RETRIEVE 198
REVISION 199
SAVE 200
SEARCH 204
SHOW 206
SMASH 207
STEP 208
STRETCH 212
TMENU 221
TNAME 222
TSLANT 224
UNDO 225
UNITS 226
VOLUMES 228
WINDOW 230
WRAP 234

Archive Reference
Information Provided 239
Archive File Format 240

Archive Commands 248-300

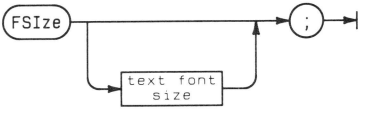
Using this Reference

This command syntax reference applies to HP EGS version 2.0. The reference contains several introductory sections and an alphabetical listing of HP EGS system-level commands. The introductory sections explain the illustrations and terminology used for the commands, the types of commands, and how to enter them.

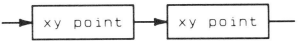
Interpreting the Syntax Diagrams

The syntax diagrams graphically explain the syntax of each command by illustrating the possible ways you can enter a command. When using the diagrams, consider the following points:

- The command syntax is connected by lines ending with directional arrows. By following these lines in the correct direction, you can construct any combination of command elements.

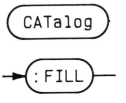


- Command elements separated by a line must have a space, comma, or (ENTER) between them. For example, a line between the two xy points indicates that coordinates must be separated by a delimiter (such as `ADD R 0,0 55,65;` or `ADD R 0,0,55,65;` or `ADD R 0,0`(ENTER)`55,65;`).

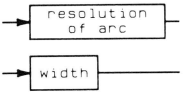


- Command elements adjacent to each other must be entered together. For example, A (for Arc) and the layer number must be entered together (such as `A6`).

- Upper-case characters in a circle or oval represent the minimum letters you need to enter. For example, to initiate CATALOG, you must enter at least `CAT` or `cat`. To specify an option such as :FILL, you must enter `:FILL` or `:fill`.



- Words and phrases contained in rectangles are command parameters. For these, you enter the appropriate characters, values, or text. The parameters for each command are explained in the table which follows the syntax diagram.



Using Component Abbreviations

The syntax diagrams often use descriptors to represent individual or collective components. These descriptors represent the following components:

Component	Abbreviations
arc	A
circle	C
dimension	D
every component	E
hatch	H
instance	I
line	L
marker	M
note (stick letter)	N
oval (or ellipse)	O
polygon	P
rectangle	R
text (block letter)	T

The following commands refer to component descriptors in their syntax diagrams:

Add	Generate	Move
Archive	Group	Show
Area	Help	Step
Copy	List	Stretch
Delete	Modify	Wrap
Gather		

Understanding the Types of Commands

HP EGS has two types of system-level commands: primary and secondary. Primary commands alter the drawing data and can be interrupted by secondary commands. However, a primary command is terminated only if interrupted by another primary command. The following commands are primary commands:

Add	Generate	Remove
Archive	Group	Retrieve
Back_annotate	List	Save
Copy	Load	Smash
Delete	Modify	Step
Edit	Move	Stretch
Exit	Nop	Undo
Gather	Quit	Wrap

Secondary commands do not alter the drawing data. You can enter a secondary command without affecting the current primary command. However, a secondary command is terminated if interrupted by *any* other command. A semicolon (;) also terminates a secondary command. The following commands are secondary commands:

Area	Dim_scale	Help	Point
Calibrate	Dim_tfsize	Identify	Prefix
Catalog	Dim_tslant	Iname	Resolution
Change	Dim_units	Input	Revision
Cursor	Display	Level	Search
Datum	Distance	Locator	Show
Decimals	Dump	Lock	Tmenu
Define	Echo	Memory	Tname
Dim_dec	Equate	Menu	Tslant
Dim_dir	Evaluate	Pack	Units
Dim_extend	Filecopy	Page	Volumes
Dim_fsize	Fsize	Pause	Window
Dim_radix	Grid	Plot	

Entering Commands

When you add components to your drawing, you must be aware of the current snapping mode and how it affects placement of the component. The following section will explain snapping modes.

Selecting Snapping Modes

Although your drawing area appears to be a solid screen, you can place points only at locations recognized by the system. These locations are determined by the snapping mode.

The snapping modes are:

:GRID — Snaps to the nearest user grid point. This is the default method.

:PRIMITIVE — Snaps to the nearest point on a primitive. A primitive is any basic component (such as a arc, circle, dimension, line, oval, polygon, rectangle, or text).

:VERTEX — Snaps to the vertex closest to the cursor. The system recognizes the following component vertices:

Component	Vertices
Line, polygon, and rectangle:	endpoint or corner.
Circle:	center
Arc:	endpoints or center.
Marker:	marker itself.
Note, text, and instance:	origin.

:INTERSECT — Snaps to either the component intersection nearest the cursor or the nearest vertex.

:RAWPOINT — Snaps to the nearest system point regardless of the current grid setting. This mode gives the finest resolution.

The advantage of snapping modes is that you can use them to specify where a point is placed. For example, if you are working on an intricate drawing and want only to place points on component vertices, you might enter `:VER` (The system requires only the first three letters of the snapping mode). Then if you use your stylus to select a point which was not exactly on a vertex, the system would snap to the nearest vertex.

You can change the snapping mode at any time. The screen displays the current mode in the bottom left corner. Grid mode, for example, would be indicated by `:GRID`.

When in primitive, intersect, or vertex modes, the size of the cursor determines the size of the area in which the system searches for points. If a valid snapping point is not found in the primitive, intersect, or vertex modes, the system snaps to the nearest user grid point.

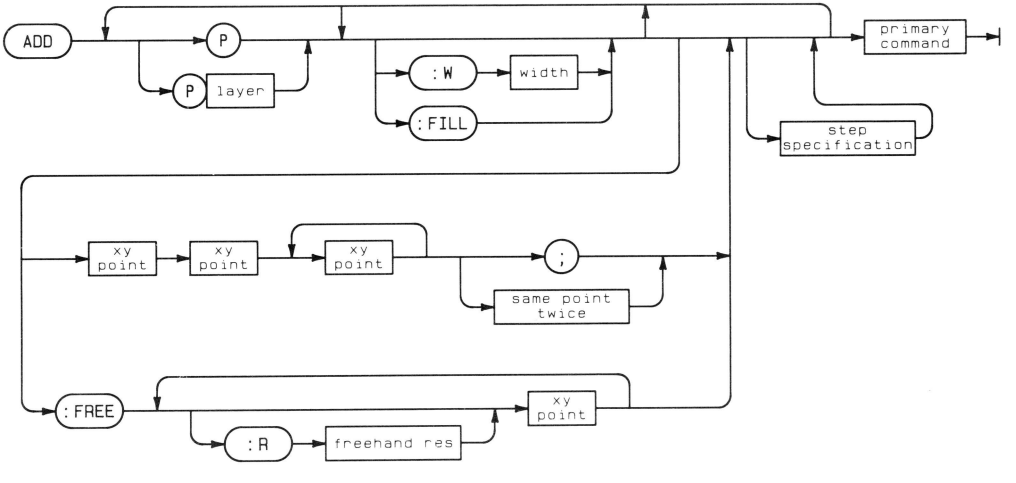
For an example of changing snapping modes while executing a command, see ADD HATCH.

Completing Primary Commands

A primary command can either be terminated or completed. You can *terminate* a primary command by entering another primary command. The way you *complete* a primary command depends on whether it requires a fixed or variable number of points.

- Commands which require the entry of a *fixed* number of points are completed when the last piece of data is entered. For example, with ADD RECTANGLE you complete the command by entering the required two points.
- Commands which require a *variable* number of points are completed by either adding a semicolon (;) or by selecting the last point twice. For example, with ADD POLYGON you can enter multiple points until completing the polygon with a semicolon or by selecting the last point twice.

Many primary commands remain active after completion. An active command allows entry of certain elements without re-entering the complete command string. The syntax diagrams illustrate active command elements by returning arrows. For example, look at the ADD POLYGON syntax diagram:



After you enter three or more xy points and a semicolon, the command is completed. However, because it is still active you could enter three or more xy points and a semicolon repeatedly without re-entering ADD or changing the width or :FILL options.

Some primary commands do not remain active after completion. These commands are:

Archive	List	Remove
Edit	Load	Retrieve
Exit	Nop	Save
Generate	Quit	Undo

Using the HP EGS Module Commands

This section explains some of the differences between system-level commands and the HP EGS module commands. These commands may occasionally confuse you until you recognize them.

Unlike the system-level commands in this reference, HP EGS module commands use macros. A macro is a one-word term which represents a string of one or more system-level commands or other macros. An example of a macro is ADD* in the HP EGS Engineering Graphics Module. Although it appears similar to the system-level command ADD, ADD* consists of several commands and other macros. Therefore, the system performs several steps each time you select ADD* from the screen menu. Because of this, macros have the obvious advantage of saving the time and effort needed for repetitive command entry.

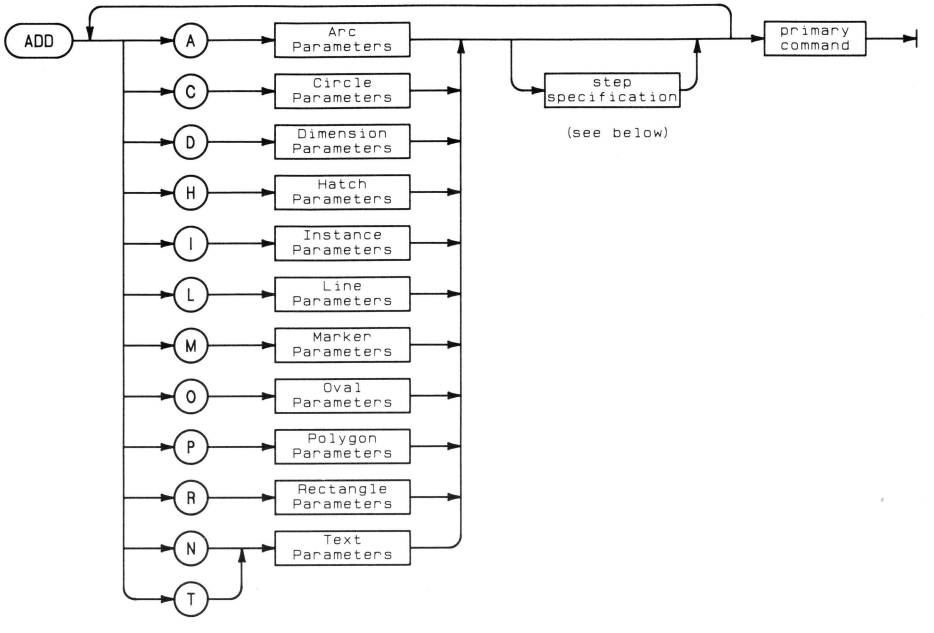
You should keep two things in mind when using both system-level commands and the module macros:

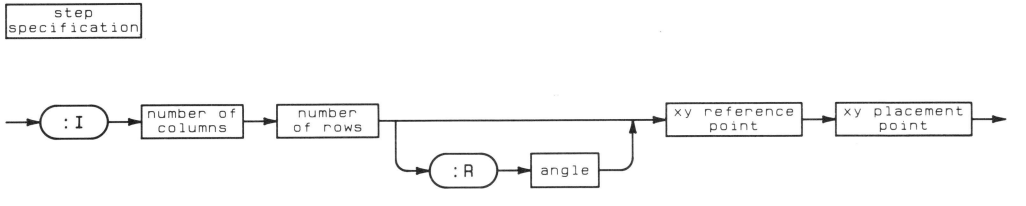
1. Some of the parameters (such as font size, width, and snapping mode) that you enter with system-level commands are not recognized by the macros of the HP EGS modules. This is because many of the macros rely on previously set macros for these parameters.
2. The following macros do nothing but display their options when selected from one of the HP EGS module screen menus.

INFO*	SAVE*
OUT*	SHOW*
RULE*	WIND*

ADD

The ADD command is a unique command because it must be accompanied by a component descriptor. However, once you enter ADD, you can add an unlimited number of components without re-entering ADD each time. ADD is a primary command.

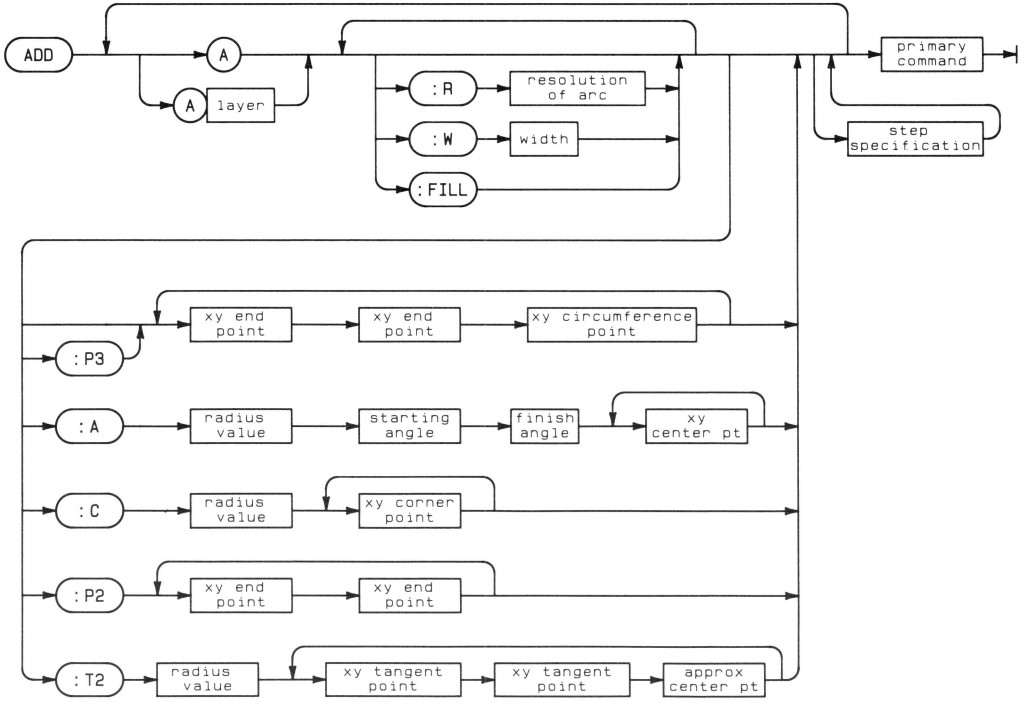




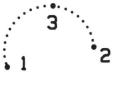
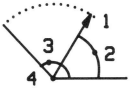
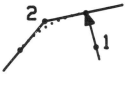
Item	Description/Default	Range Restrictions
Commands		
A	A descriptor specifying an arc component.	
C	A descriptor specifying a circle component.	
D	A descriptor specifying a dimension component.	
H	A descriptor specifying a hatch component.	
I	A descriptor specifying an instance component.	
L	A descriptor specifying a line component.	
M	A descriptor specifying a marker component.	
O	A descriptor specifying an oval (ellipse) component.	
P	A descriptor specifying a polygon component.	
R	A descriptor specifying a rectangle component.	
N	A descriptor specifying a note component.	
T	A descriptor specifying a text component.	
Option		
:I	Specifies to add an implicit step to the component. An implicit step multiplies the component by the array you enter. Implicitly added components act as a group. Therefore, any changes you make on one of the components of the array will be reflected by all of the array components. (See ADD RECTANGLE for an implicit step example.)	
Parameters		
number of columns	The number of columns in the array.	
number of rows	The number of rows in the array.	
:R	Specifies to rotate the array.	
angle	The angle of rotation of the array. The angle is measured from an imaginary horizontal line.	-360° to +360°
xy reference point	A location which acts as a reference point.	
xy placement point	A location which defines the distances between the reference points of the components of the array.	

ADD ARC

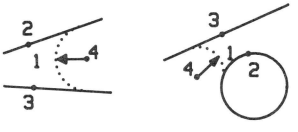
ADD ARC adds an arc component to your drawing. If you add an arc with the isometric grid turned on, the arc is represented by a partial oval. A partial oval can be added by any of the five arc methods. ADD ARC is a primary command.



Item	Description/Default	Range Restrictions
Command		
A	The descriptor specifying an arc component.	
layer	Layer to which you are adding the arc. The default is the last layer entered.	1-255

Item	Description/Default	Range Restrictions
Options		
:R resolution of the arc	Specifies the angle (in degrees) between endpoints of the line segments which form the arc. 1° is the smallest resolution that will have line segments. A 0° resolution will draw a smooth arc. The default is defined by the RESOLUTION command.	0-120°
:W width	Specifies the width of the arc outline.	Less than the diameter.
:FILL	Specifies to fill the outline of the arc. :FILL is ignored if the width is zero.	
step specification	Adds an implicit step to the component. See ADD for explanation.	
Methods		
:P3	Enter the two endpoints of the arc (1,2) and a point on its circumference (3). This is the default method. 	
:A	Enter the radius of the arc (1). Enter the angle (2) from an imaginary horizontal line counterclockwise to the beginning of the arc. This is the starting angle. Enter the angle (3) from the imaginary horizontal line counterclockwise to the end of the arc. This is the finish angle. Enter the center point of the arc (4). The arc extends between the two angles. 	
:C	Draws a fillet (smooth corner) with an arc. Enter the radius of the arc (1) and enter or select the vertex of the corner you wish to smooth (2). The vertex can be formed by segments of lines, rectangles, or polygons. The vertex must be on the endpoints of two segments. 	

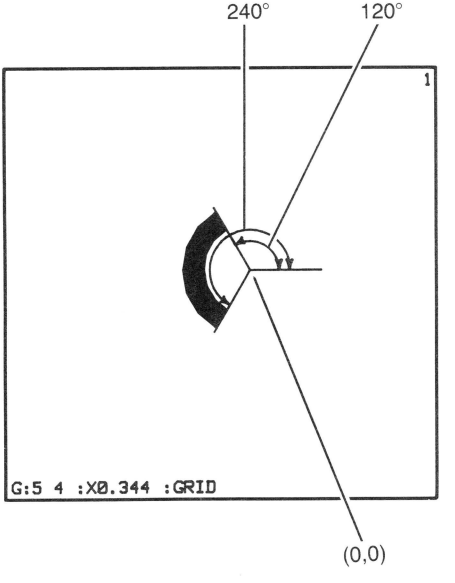
(Table continued)

Item	Description/Default	Range Restrictions
:P2	Enter the two endpoints of the arc (1,2). A semi-circle is drawn in a counter-clockwise direction from the first endpoint to the second.	
:T2	Enter the radius value of the arc (1), two points approximately tangent to a component (2,3) and the approximate center point of the arc (4). The arc is drawn between the two tangent points. 	
Parameters		
xy endpoint	Endpoint of the arc.	
xy circumference point	A point on the circumference of the arc.	
radius value	Radius of arc.	
starting angle	Angle (in degrees) measured counterclockwise from an imaginary horizontal line. This will be the starting point of a an arc drawn by the :A method.	-360° to +360°
finish angle	Angle (in degrees) measured counterclockwise from an imaginary horizontal line. This will be the endpoint of an arc drawn by the :A method.	-360° to +360°
xy center point	Center of arc.	
xy corner point	Defines a vertex formed by two intersecting line segments.	
xy tangent point	Defines a point on a component to which the arc is tangent.	
approximate center point	The estimated center point for an arc that is tangent to two components. The system calculates the true center.	

Example

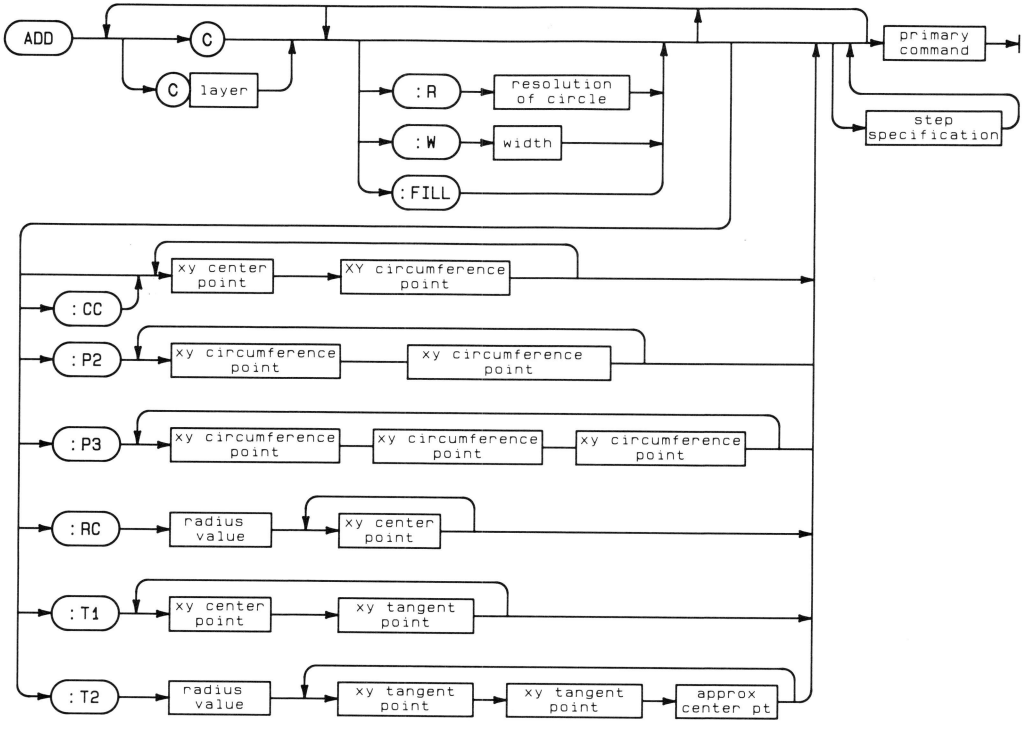
```
ADD A4 :R 20 :W 20 :FILL :A 50 120 240 0,0;
```

This example adds an arc on layer 4 . The arc has a resolution of 20° and has a filled width of 20. The radius is 50, its starting angle 120°, its finish angle 240°, and its center point is (0,0).



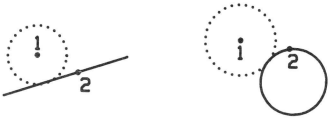
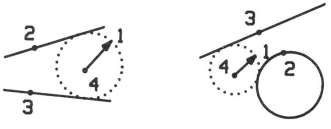
ADD CIRCLE

ADD CIRCLE adds a circle to your drawing. If you add a circle with the isometric grid turned on, the circle is represented by an oval. An oval can be added by any of the six circle methods. ADD CIRCLE is a primary command.



Item	Description/Default	Range Restrictions
Command		
C	A descriptor specifying a circle component.	
layer	Layer on which you are adding the circle. The default is the last layer entered.	1-255
Options		
:R resolution of circle	Specifies the angle (in degrees) between endpoints of the line segments which form the circle. 1° is the smallest resolution that will have line segments. A 0° resolution will draw a smooth circle. The default is defined by the RESOLUTION command.	0-120°
:W width	Specifies the width of the circle.	Less than the diameter.
:FILL	Specifies to fill the circle outline. If you have specified a width, only the width will be filled.	
step specification	Adds an implicit step to the component. See ADD for explanation.	
Methods		
:CC	Enter the center point (1) and a point on the circumference of the circle (2). This is the default method.	
:P2	Enter the two endpoints of the diameter of the circle (1,2).	
:P3	Enter three points which define the circumference of the circle (1,2,3).	

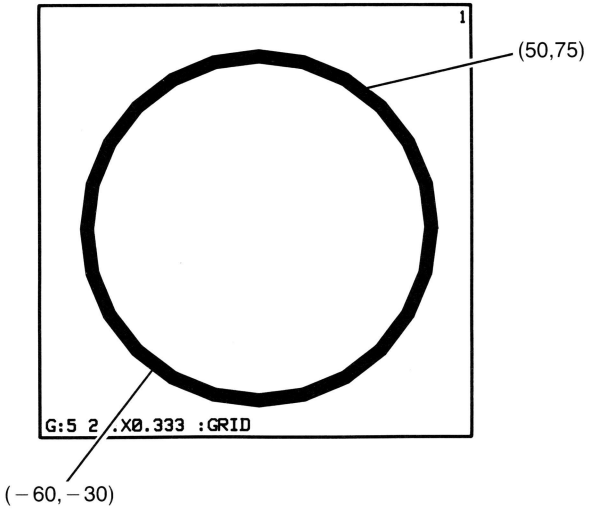
(Table continued)

Item	Description/Default	Range Restrictions
:RC	Enter the radius (1) and the center point (2).	
:T1	Enter the center point (1) and a point approximately tangent to a component. 	
:T2	Enter the radius (1), points approximately tangent to two components (2,3), and the approximate center point (4). 	
Parameters		
xy center point	Center point of the circle.	
xy circumference point	Point on circumference of the circle.	
radius value	Radius of circle.	
xy tangent point	A point on a component to which the circle is to be tangent.	
approx center pt	The approximate center point for a circle that is tangent to two components. The system will calculate the true center.	

Example

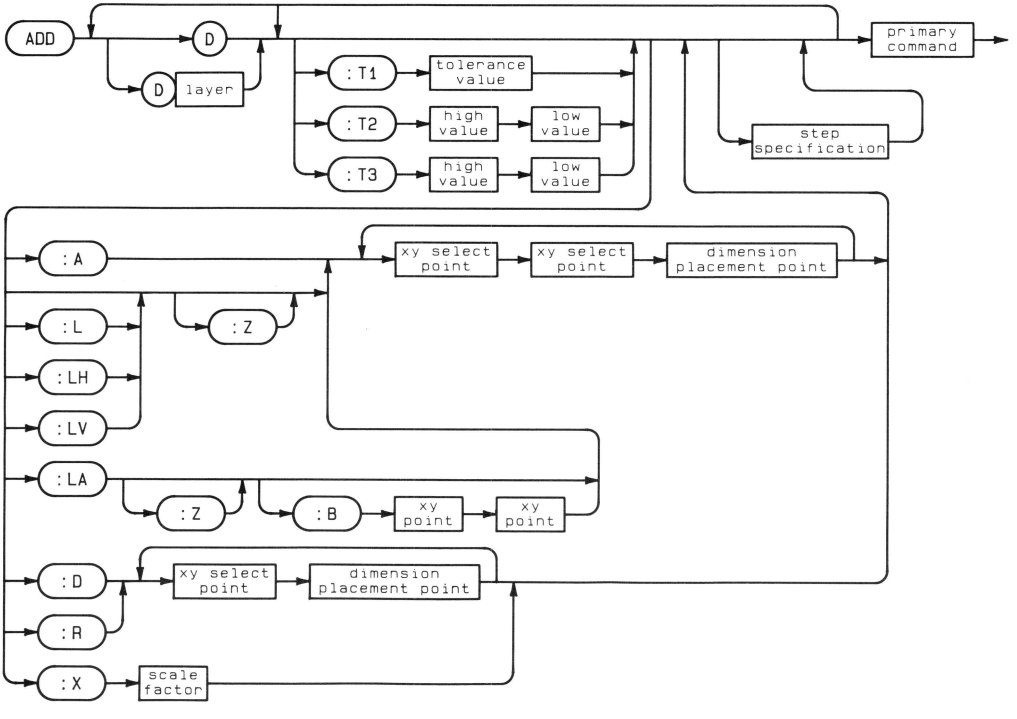
```
ADD C2 :R 15 :W 10 :FILL :P2 50,75 -60,-30 ;
```

This example adds a circle on layer 2. The circle has a resolution of 15° and a filled width of 10. Its diameter is the line between the points (50,75) and (-60,-30). The illustration below shows the circle as it would appear on your screen.



ADD DIMENSION

ADD DIMENSION adds dimension lines and values to your drawing. ADD DIMENSION is a primary command.



Item	Description/Default	Range Restrictions
Command		
D	A descriptor specifying a dimension component.	
layer	The layer number on which the dimension is added. The default is the last layer entered.	1-255
Options		
:T1 tolerance value	Specifies symmetric tolerancing about the dimension by the given tolerance value. The dimension value is followed by a "plus/minus" tolerance value. (See example.)	
:T2 high/low value	Specifies nonsymmetric tolerancing about the dimension by the given high value and low value. The dimension value is followed by a "plus" high value and a "minus" low value. (See example.)	
:T3 high/low value	Specifies nonsymmetric tolerancing about the dimension by the given high value and low value. The dimension value added to the high value lies above the dimension value added to the low value. (See example.)	
step specification	Adds an implicit step to the component. See ADD for explanation.	
Methods		
:A	Specifies an angular dimension in degrees between two line segments.	
:L	Specifies a line dimension. This is the default method.	
:LH	Specifies a horizontal line dimension.	
:LV	Specifies a vertical line dimension.	
:LA	Specifies an angular line dimension.	
:Z	Specifies to draw zero-base datum dimensions (one-arrow dimension lines). The second point determines arrowhead and extension line placement.	
:B	Specifies that the angular line dimension will be parallel to a base line designated by two xy points.	
:D	Specifies a diameter dimension of an arc or circle.	
:R	Specifies a radius dimension of an arc or circle.	

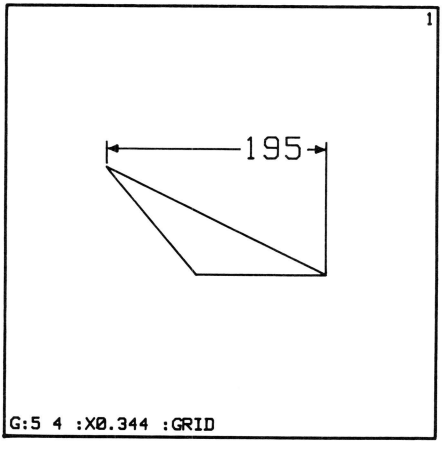
(Table continued)

Item	Description/Default	Range Restrictions
Parameters		
:X scale factor	Specifies the scale factor of the dimension to add. This scale factor remains in effect until the command is terminated by a primary command. The default is 1.	Must be greater than 0
xy point	Specifies the points which define a base line for angular line dimensions.	
xy select point	Specifies a point on an existing component.	
dimension placement point	Specifies the xy point for the location of the dimension number values. The dimension value is associated text.	

Example

```
ADD D :LA :B -60,-35 55,-35 -140,60 55,-35 25,75 ;
```

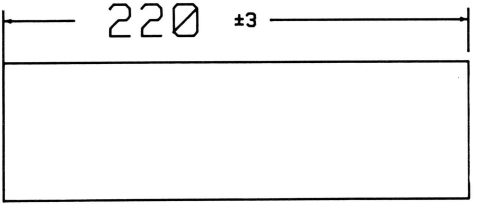
This example adds an angular line dimension to one leg of a triangle. The dimension is parallel to the base leg which has the points (-60,-35) and (55,-35). The dimension value is located at (25,75).



Example

```
ADD R -80,-30 140,35;
ADD D :T1 3 :L -80,-35 140,35 0,55;
```

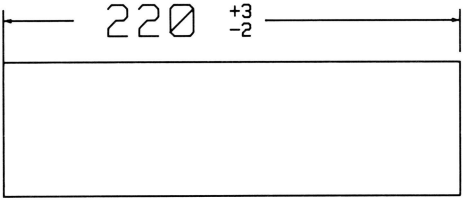
This example dimensions a rectangle with symmetric tolerancing. Note: For the following three examples, the DIM_FSIZE is 15 user units and DIM_TFSIZE is 6 user units in the HP EGS General Drawing personality.



Example

```
ADD R -80,-30 140,35;
ADD D :T2 3 2 :L -80,-30 140,35 0,55;
```

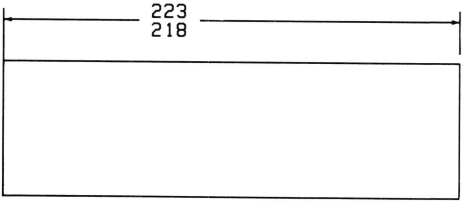
This example dimensions the rectangle with nonsymmetric tolerancing. The high tolerance value is 3 user units greater than the absolute dimension and the low tolerance value is 2 user units less than the absolute dimension.



Example

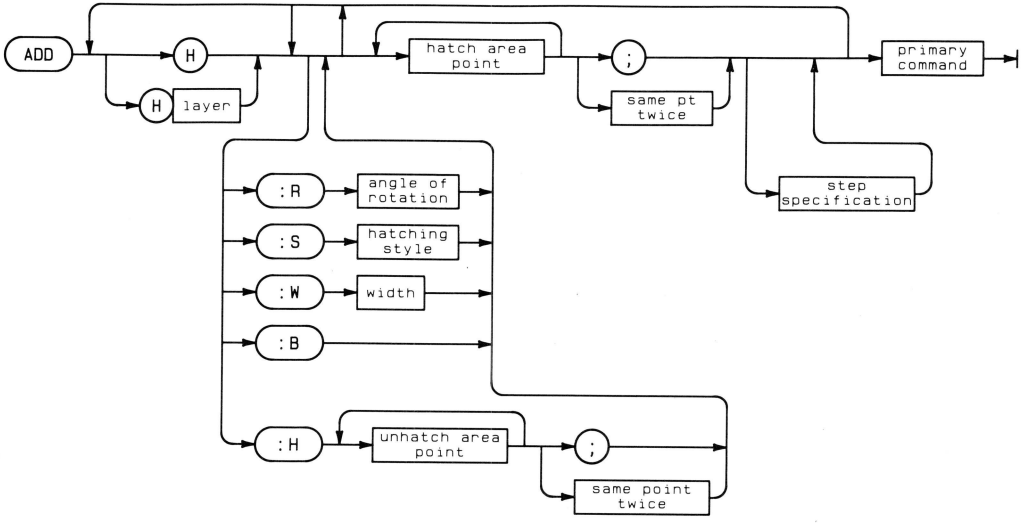
```
ADD R -80,-30 140,35;
ADD D :T3 3 2 :L -80,-30 140,35 0,55;
```

This example also specifies nonsymmetric tolerancing but the high dimension value is placed above the low dimension value.



ADD HATCH

ADD HATCH adds a hatched region to your drawing. You specify the boundary of the hatched region and the boundaries of any unhatched areas within the region. ADD HATCH is a primary command.



Item	Description/Default	Range Restrictions
Command		
H	The descriptor specifying hatching.	
layer	The layer on which you are adding the hatching. The default is the last layer entered.	1-255
Options		
:R angle of rotation	Angle in degrees between the hatching and the horizontal. The default is 0°.	-90° through +90°
:S hatching style	Style of hatching. Specify by one of the following numbers: 1 solid fill (default) 2 single hatching 3 cross-hatching If you specify style 2 or 3, you must also specify a width or the hatched area will appear solid.	1-3
:W width	Width in user units between hatching lines . The default is 0.	
:B	Specifies to draw the boundary of the hatched region.	
:H	Option to define unhatched holes in the hatched polygon. :H must precede the specification of each unhatched area.	
step specification	Adds an implicit step to the component. See ADD for explanation.	
Parameters		
unhatch area point	Point on the boundary of unhatched region.	Maximum 1000 points
hatch area point	Point on the boundary of hatched region.	Maximum 1000 points

Hints

You can select the boundaries of hatched and unhatched regions while in primitive mode or one of the other snapping modes. `:H` must precede designation of each unhatched region.

- Primitive mode: Use this mode when one primitive (circle, line, oval, polygon, or text) forms the entire boundary. If a component with width is selected, the width parameter is ignored.

  Set the snapping mode to `:PRI` and select the component. Once selected, the component becomes a boundary in one of the following ways:

circle/oval	The circle/oval is converted to a regular polygon using the resolution specified when it was drawn. Notice that the hatching will not exactly match the circle if the resolution is changed during plotting.
polygon	The polygon becomes the boundary of the region.
rectangle	The rectangle becomes the boundary of the region.
text	The bounding rectangle surrounding the text becomes the boundary of the region. Notice that this would generally be used to form the boundary of an unhatched region, as it would not be practical to hatch text.
all other components	Selection of all other primitives is ignored.

- Other snapping modes: Use these other modes to define the hatched or unhatched boundaries one point at a time.

  You may find it advantageous to switch to appropriate snapping modes as you go along. For instance, the grid, vertex, and intersection modes may be used to select straight line portions of the boundary. When you encounter a portion of an arc, circle or oval,:

  1. Select the endpoint of the arc while in vertex or intersection mode.
  2. Switch to the primitive mode and select a point along the outline of the arc.
  3. Return to the vertex or intersection mode to select the other endpoint of the arc.

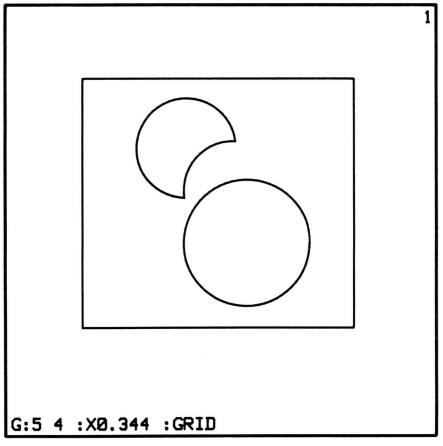
Example

The following example illustrates hatching while using different snapping modes. The example will use keyboard commands (although you could use the stylus to select the coordinates and snapping modes) to:

1. Create a simple drawing,
2. Outline the unhatched regions, and
3. Outline the hatched region.

1. Create the drawing below by entering the following commands:

```
ADD C 10,-70 35,-120;
ADD R -135,-145 105,75;
ADD A -45,-30 0,20 -5,35;
ADD A -45,-30 0,20 -30,10;
```



2. Define the circle as an unhatched region by entering:

```
ADD H :R45 :S2 :W5 :PRI :H 35,-120
```

The hatching rotation is 45°, the style is 2, and the width is 5 user units. Notice this step uses the primitive snapping mode and only one point (35,-120) because the unhatched region, a circle, is a single component. `:H` specifies the beginning of an unhatched region.

(Example continued)

Define the two arcs as a second unhatched region by entering:

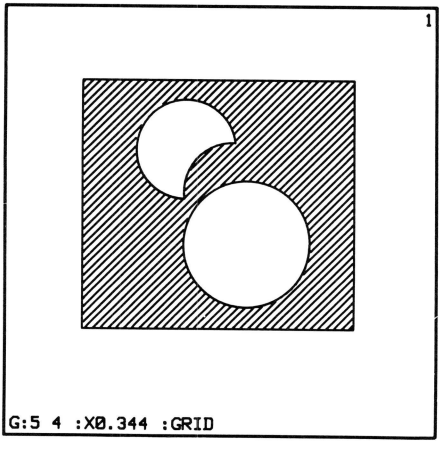
```
:H :VER  -45,-30 :PRI  -30,10 :VER  0,20 :PRI  -5,35 :VER  -45,-30  -45,-30
```

This step begins defining the second unhatched region with :H. Since the region consists of more than one component, the points are defined by alternating between primitive and vertex snapping modes. Notice that when a boundary consists of more than one component, you must re-enter the last point (in this case, -45,-30) or enter a semicolon to complete the definition of the second unhatched area.

3. Define the rectangle as the hatched region by entering:

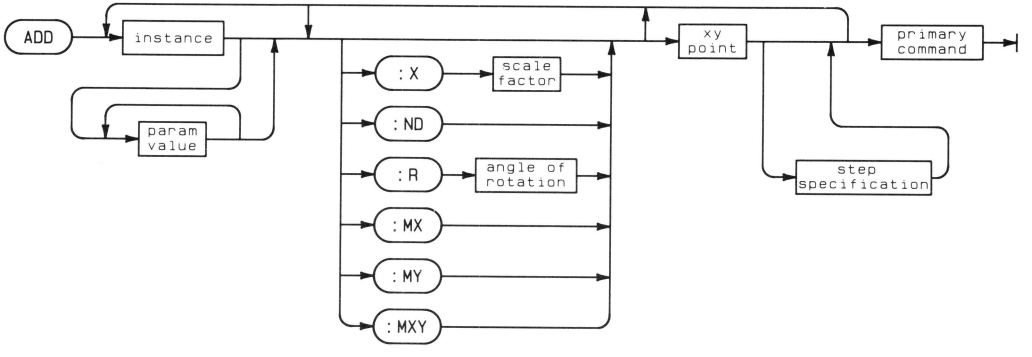
```
:PRI  105,75  ;
```

This step completes the hatching by defining the hatched boundary. As in step 2 the boundary is a single component and thus can be defined with one point (105,75) while in primitive snapping mode.



ADD instance

The ADD command, when followed by a drawing file name, adds an instance to your drawing. ADD instance is a primary command.



Item	Description/Default	Range Restrictions
Command		
instance	The full file name of an instance. This can be an graphical instance such as a drawing file stored with the suffix _d, or a macro instance created with DEFINE :I. A macro instance is stored as a text file with the suffix _i.	
Options		
:X scale factor	Specifies how to change the scale of the instance when it is displayed with your drawing. This factor acts as a multiplier so, for example, 0.5 specifies to display the instance one-half of its original size. The default is 1.	
:ND	Specifies not to add text associated with the instance drawing. :ND is primarily used when the drawing is to be archived.	
:R angle of rotation	Angle in degrees to rotate the instance when it is added to the drawing.	-360° to +360°
:MX	Specifies that the instance added to the drawing be mirrored about the X axis.	
:MY	Specifies that the instance added to the drawing be mirrored about the Y axis.	
:MXY	Specifies that the instance added to the drawing be mirrored about both the X and Y axes.	
step specification	Adds an implicit step to the instance. See ADD for explanation.	
Parameters		
param value	Parameter values for a macro instance.	
xy point	Point where the instance is to be added.	

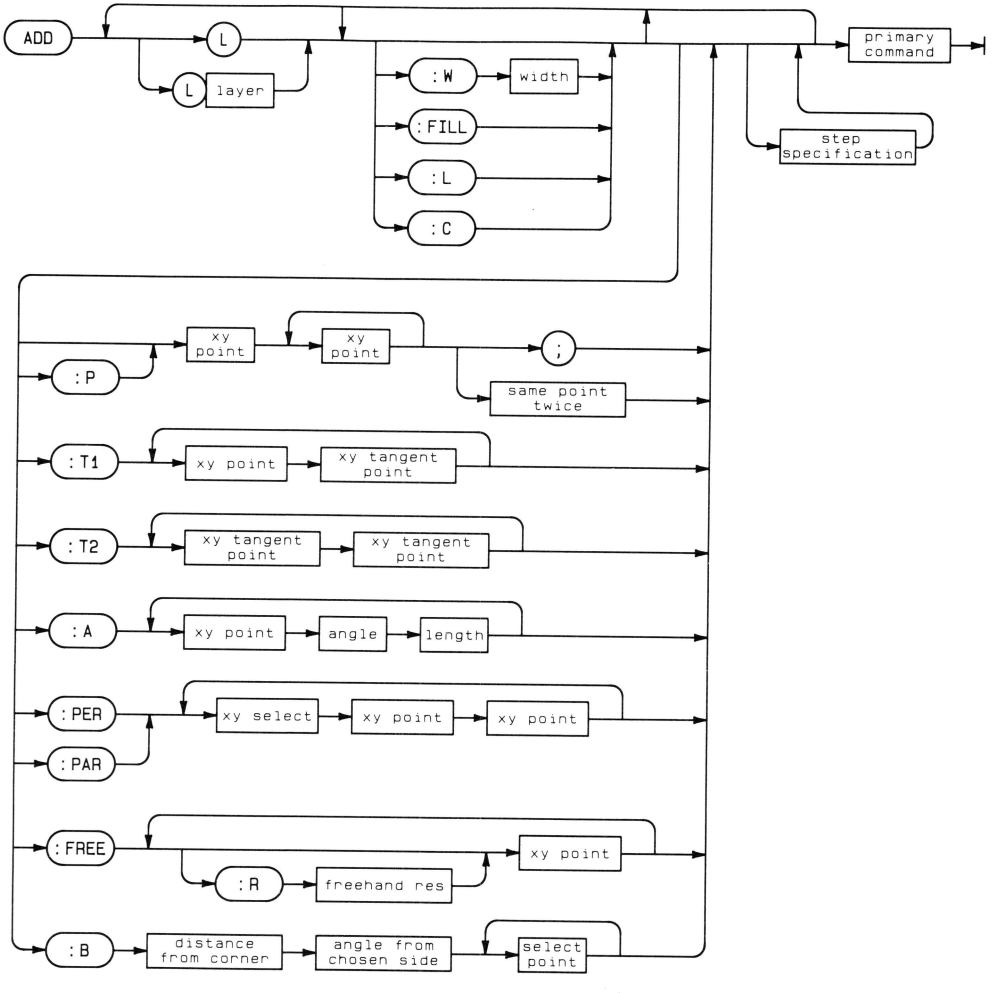
Example

```
ADD #5:/HP_EGS/EWDEMO/hub_d :MX 100,200
```

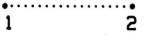
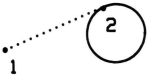
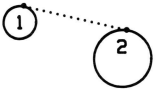
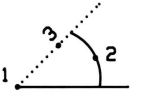
This example adds the instance called hub_d to your drawing. This instance is in the volume #5:/HP_EGS/EWDEMO. When drawn on the screen, hub_d will be mirrored about the X axis and has its origin at (100,200).

ADD LINE

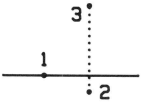
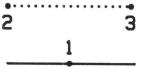
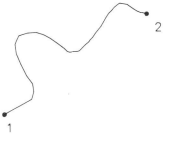
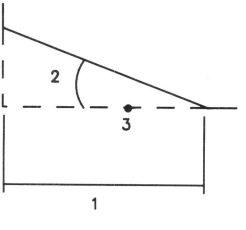
ADD LINE adds a line component to your drawing. ADD LINE is a primary command.



Item	Description/Default	Range Restrictions
Command		
L	A descriptor specifying a line component.	
layer	Layer on which you are adding the line. The default is the last layer entered.	1-255

Item	Description/Default	Range Restrictions
Options		
:W width	Specifies the width of the line.	
:FILL	Specifies to fill the line if a width has been specified.	
:L	Specifies to draw a leader line with the arrowhead at the first endpoint.	
:C	Specifies a line of infinite length (this is a construction line).	
step specification	Adds an implicit step to the component. See ADD for explanation.	
Methods		
:P	Enter the endpoints (1,2) of the line. This is the default method. Up to 255 points can be entered with this method. 	
:T1	Enter a point (1) and a point approximately tangent to a circle or arc (2). 	
:T2	Enter two points (1,2) approximately tangent to circles and/or arcs. 	
:A	Enter one of the endpoints of the line (1). Enter the angle (2) drawn counterclockwise from an imaginary horizontal line to the new line. Enter the length (3) of the line. This method always uses the rawpoint snapping mode. 	

(Table continued)

Item	Description/Default	Range Restrictions
Methods (cont)		
:PER	Enter a point (1) on or near a line you wish to be perpendicular to your new line. Enter the two endpoints (2,3) of your new line. 	
:PAR	Enter a point (1) on or near a line you wish to be parallel to your new line. Enter the two endpoints (2,3) of the new line. 	
:FREE	Specifies to draw a free hand line with the graphics input device (such as a stylus). Select or enter the first point (1) and then use the input device to trace the line on the tablet. Select a second point (2) or move the input device out of the current viewport to terminate the line. 	
:B	Specifies to add a bevel or chamfer to a corner. Enter the distance from the corner (1), angle from the chosen side (2), and select a point (3) on one of the segments of the corner. Point 3 determines which side distance 1 and angle 2 affects. 	

(Table continued)

Item	Description/Default	Range Restrictions
Parameters		
xy point	Endpoint of the line.	
xy tangent point	Point on an arc or circle to which the line is tangent.	
angle	The angle measured counterclockwise from an imaginary horizontal line to the line you are drawing.	
length	Length of line.	
xy select	Point on or near an existing line which will be perpendicular or parallel to your new line.	
:R freehand res	Specifies the resolution in user units between freehand line points. The default is 0. The system places a freehand line point when the stylus has traveled the resolution length or in the x or y direction a greater distance than this length. A resolution of 0 is the maximum system resolution.	0≤res≤screen width
distance from corner	The distance from a corner to the beginning of a chamfer.	
angle from chosen side	The angle of cut for the chamfer relative to the side chosen by the select point. If zero is entered then a symmetric chamfer is created. In a symmetric chamfer the distance from the corner is the same on each side.	0-90°
select point	Point on one of the edges to be chamfered. This edge is the reference for the chamfer angle when it is not 0°.	

Example

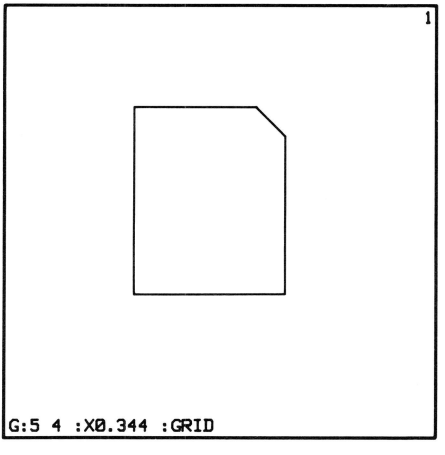
```
ADD L2  :W 10 :FILL  150,100 120,340 ;
```

This example adds a line on layer 2 with a filled width of 10. Its endpoints are (150,100) and (120,340).

Example

```
ADD R 95,120 -10,-10;
ADD L :B 20 45 95,120;
```

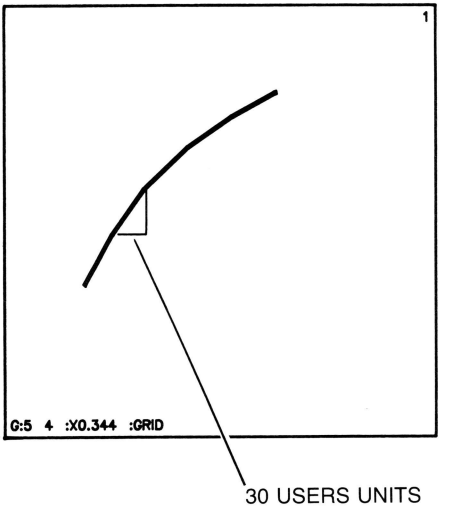
This example first adds a rectangle with corner points of (95,120) and (-10,-10). The second command adds a chamfer which begins 20 user units from the corner (95,120). The chamfer angle is 45° relative to the edge which passes through (95,120). Note: 0° always produces a symmetrical chamfer angle. 45° produces a symmetrical chamfer angle only on 90° corners.



Example

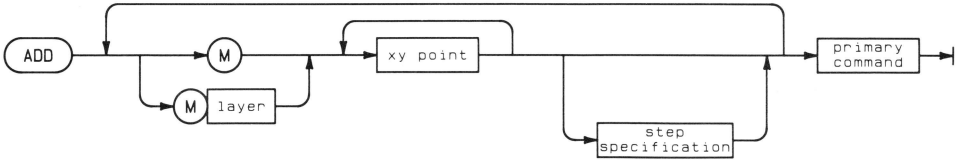
```
ADD L4 :W 5 :FILL :FREE :R 30
```

This example allows you to draw a freehand line on layer 4. The line has a filled width of 5 and a resolution of 30 user units. With this resolution, the system places a point when the stylus has traveled 30 user units in the x or y direction. In this case, the stylus moved thirty user units in the x direction when a point was placed. NOTE: FREEHAND lines and polygons have limited width and fill capabilities.



ADD MARKER

ADD MARKER adds a marker to your drawing. Markers are usually used to imply connected components on electrical schematics. The size of a marker is independent of the window magnification. ADD MARKER is a primary command.



Item	Description/Default	Range Restrictions
Command		
M	A descriptor specifying a marker component.	
layer	Layer on which you are adding the marker. The default is the last layer number entered.	1-255
Option		
step specification	Adds an implicit step to the marker. See ADD for explanation.	
Parameter		
xy point	Point where the marker is to be added.	

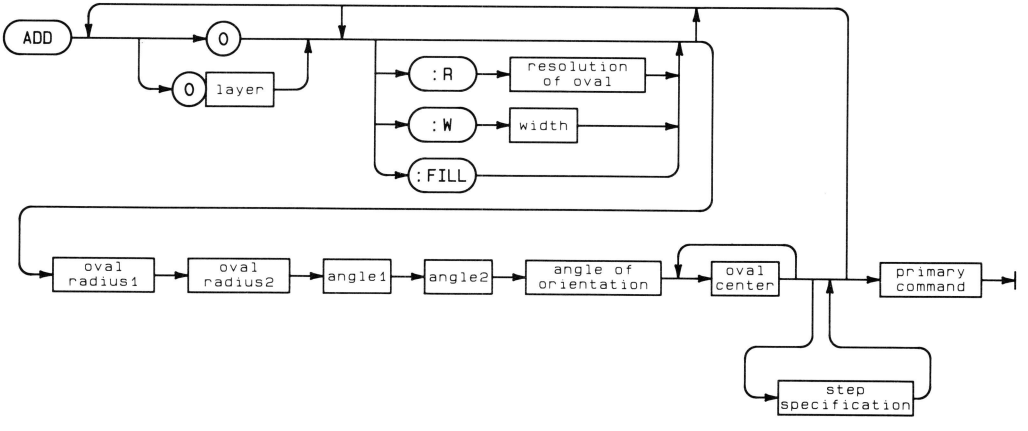
Example

```
ADD M3 20,60 ;
```

This example adds a marker on layer 3 at (20,60).

ADD OVAL

ADD OVAL adds an oval (ellipse) to your drawing. ADD OVAL is a primary command.

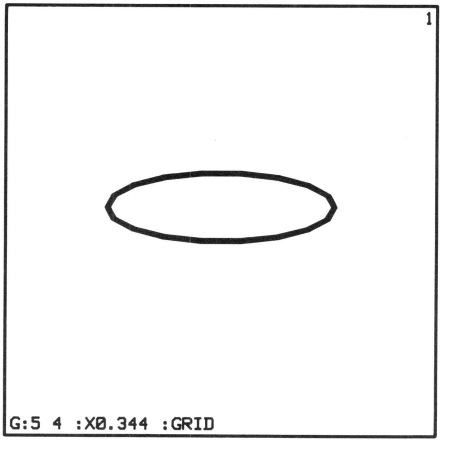


Item	Description/Default	Range Restrictions
Command		
O	A descriptor specifying an oval component.	
layer	Layer on which you are adding the oval. The default is the last layer number entered.	1-255
Options		
:R resolution of oval	Specifies the angle (in degrees) between endpoints of the line segments that approximate the oval. The default is set by the RESOLUTION command.	0-120°
:W width	Specifies the width of the oval outline.	less than the minor diameter
:FILL	Specifies to fill an oval with width.	
step specification	Adds an implicit step to the oval. See ADD for explanation.	
Parameters		
oval radius1	The length of the major radius.	
oval radius2	The length of the minor radius.	less than radius1
angle1	The angle measured counterclockwise from the major axis of the oval to the beginning point of the oval.	0-360°
angle2	The angle measured counterclockwise from the major axix of the oval to the ending point of the oval.	-360° to +360°. (must be more than angle1)
angle of orientation	The angle of rotation of the major axis measured counterclockwise from an imaginary horizontal line.	-360° to +360°.
oval center	Center point of the oval.	

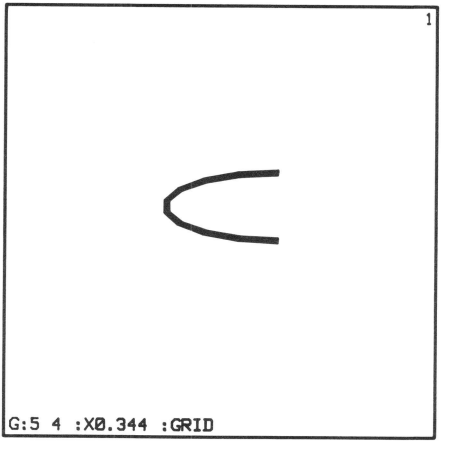
Example

```
ADD 04 :W 10  :FILL :R20 100 30 0 360 0 0,0;
```

This example adds an oval on layer 4. It has a filled width of 10, resolution of 20, major axis radius of 100 and a minor axis radius of 30. It is a complete oval since its starting angle (0°) is the same as the finish angle (360°). The center of the oval is (0,0).

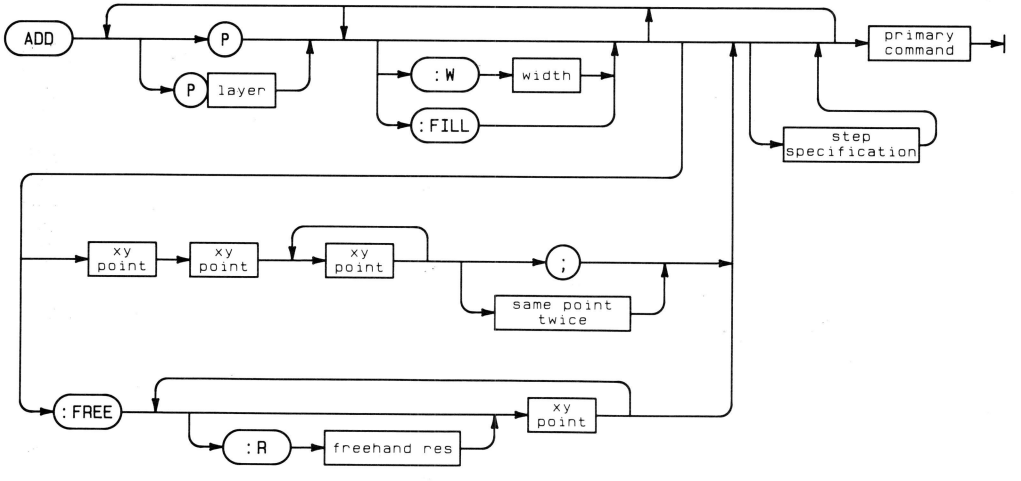


If the starting angle was 90° and the finish angle 270°, the oval would appear as below:



ADD POLYGON

ADD POLYGON adds a polygon to your drawing. A polygon could be considered to be a three or more point line which the system closes for you. ADD POLYGON is a primary command.



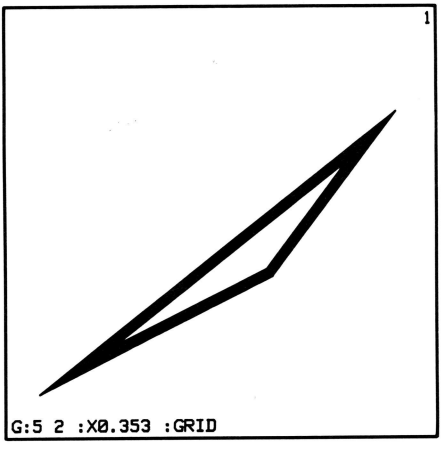
Item	Description/Default	Range Restrictions
Command		
P	A descriptor specifying a polygon component.	
layer	Layer on which you are adding the polygon. The default is the last layer entered.	1-255
Options		
:W width	Specifies the width of the polygon.	
:FILL	Specifies to fill the polygon. If you specify width, only the outline will be filled.	
step specification	Adds an implicit step to the polygon. See ADD for explanation.	

Item	Description/Default	Range Restrictions
Methods		
default method	A polygon can be drawn by selecting three or more points	
:FREE	Specifies to draw a free hand polygon. Select or enter the first point and trace the polygon with the graphics input device (such as a stylus). Select a second point or move the input device out of the current viewport to complete the polygon. Freehand polygons have limited width and fill capabilities.	
Parameters		
xy point	Location of a polygon vertex.	
freehand res	Specifies the length in user units of the line segments which make up a freehand polygon. The default is 1 user unit. The system places a freehand polygon point when the stylus has traveled the resolution length or in the x or y direction a greater distance than this length.	0<res<screen width.

Example

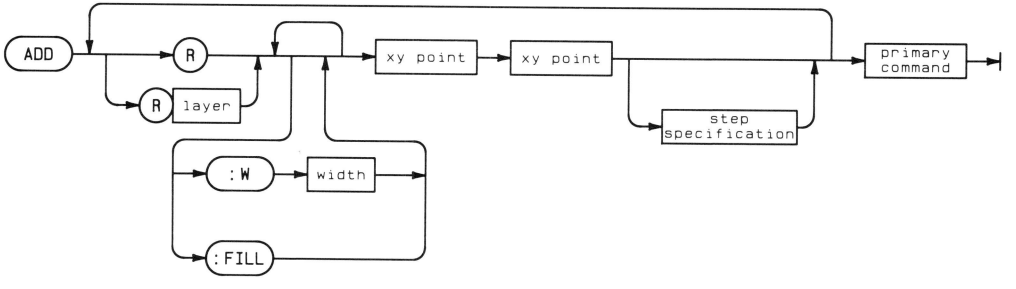
```
ADD P4 :W 10 :FILL -140,-130 120,75 30,-45;
```

This example adds a polygon to layer 4. Its outline has a width of 10 and is filled. The vertices are (-140,-130), (120,75), and (30,-45).



ADD RECTANGLE

ADD RECTANGLE adds a rectangle to your drawing. In isometric mode, rectangles are added as regular polygons. ADD RECTANGLE is a primary command.



Item	Description/Default	Range Restrictions
Command		
R	A descriptor specifying a rectangle component.	
layer	Layer on which you are adding the rectangle. The default is the last layer entered.	1-255
Options		
:W width	Specifies the width of the recangle.	
:FILL	Specifies to fill the rectangle.	
step specification	Adds an implicit step to the rectangle. See ADD for explanation.	
Parameter		
xy point	Location of opposite corners of the rectangle.	

Example

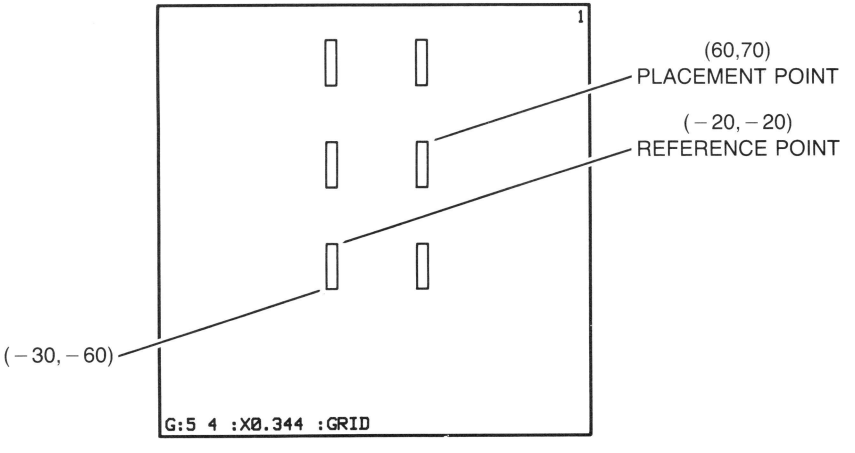
```
ADD R7  :W 10 :FILL 0,0 200,200 ;
```

This example adds a rectangle on layer 7. It has a filled width of 10 and has its two opposing corners at (0,0) and (200,200).

Example

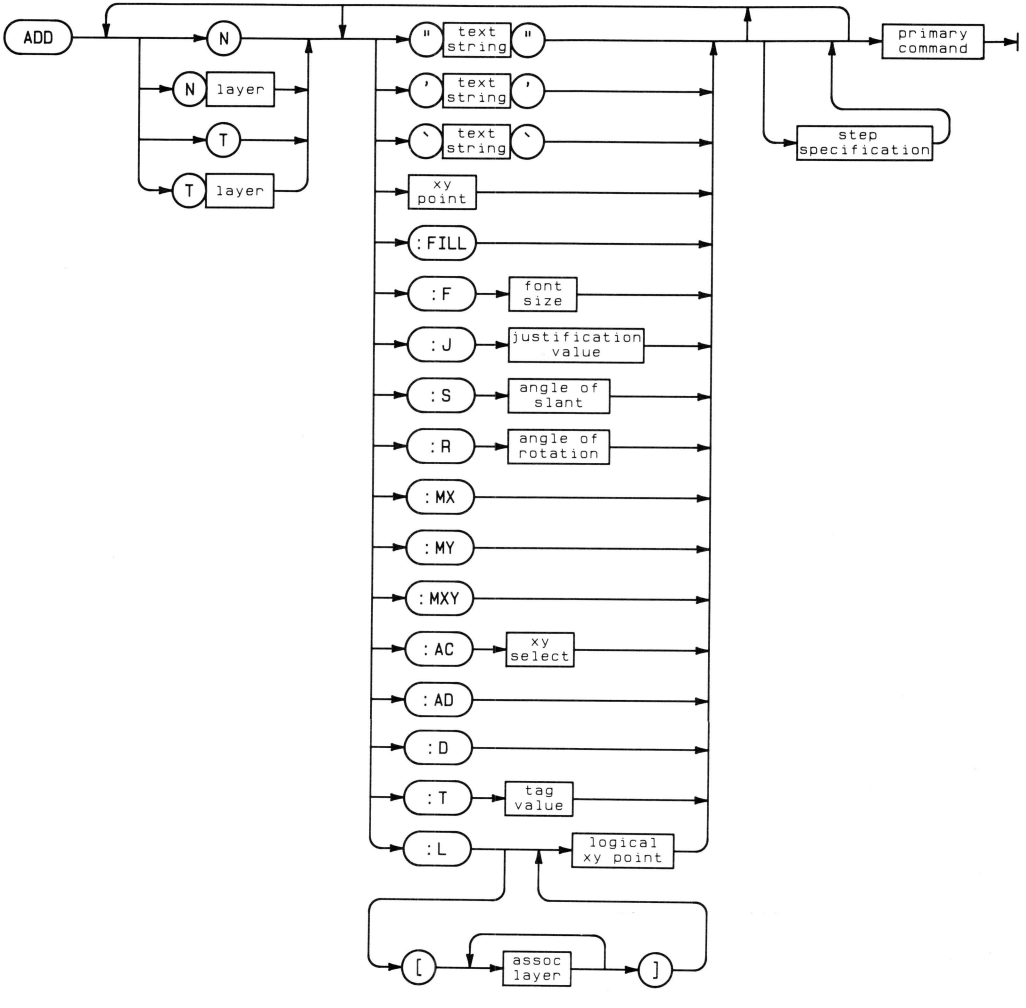
```
ADD R  -20,-20  -30,-60 :I 2 3  -20,-20 60,70;
```

This example adds a rectangle with opposing corners at (-20,-20) and (-30,-60). A step specification with 2 columns and 3 rows uses (-20,-20) as a reference point and (60,70) as the placement point. The screen appears as follows:



ADD TEXT or NOTE

ADD TEXT adds block characters that can be filled and ADD Note adds stick characters to your drawing. Both of these commands are primary commands.



Item	Description/Default	Range Restrictions
Commands		
N	A descriptor specifying a stick character.	
T	A descriptor specifying a block character. Although you may enter any characters, text characters will always display as upper case letters.	
layer	Layer on which to add the text. The default is the last layer entered.	1-255
text string	Text to be added.	
Options		
:FILL	Specifies to fill text characters . Note characters cannot be filled.	
:F font size	Height of the characters in user units.	
:J justification value	Specifies the orientation of the block of characters relative to the xy point. The default value is 1 which locates the xy point in the lower left hand corner of the boundary rectangle. (See Hints section for an illustration.)	1-9
:S angle of slant	Angle in degrees to slant the characters. The angle is measured from an imaginary vertical line (clockwise is positive).	-75° to +75°
:R angle of rotation	Angle in degrees to rotate the text. The angle is measured counterclockwise from an imaginary horizontal line.	-360° to +360°
:MX	Specifies to mirror the character about the X axis.	
:MY	Specifies to mirror the characters about the Y axis.	
:MXY	Specifies to mirror the characters about both the X and Y axis.	

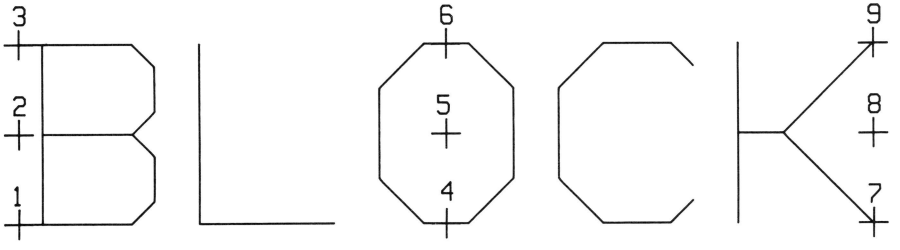
(Table continued)

Item	Description/Default	Range Restrictions
Options (cont.)		
:AC	Specifies to associate the characters with a component. You must select the particular component.	
:AD	Specifies to associate characters with the drawing being edited. If you add this drawing to another drawing as an instance, the :AD characters will be copied to :AC characters which are associated to the instance..	
:D[1]	Specifies to display the text with the drawing when it is added as an instance. The default is to not display text.	
:T[1]	Specifies to include a tag value with the associated text. This is used in post-processing applications to specify a type or class of associated text. Tag values 1000-1999 are HP-reserved.	
:L[1]	Specifies to include a logical location for the associated text.	
step specification	Adds an implicit step to the component. See ADD for explanation.	
Parameters		
xy point	Point to add the text. This can be entered or selected with the stylus.	
xy select	Location of the component to associate characters.	
tag value[1]	Tag value of the associated text.	-32768 to 32767
logical xy point[1]	Selects the logical location of the associated text.	
associated layer[1]	Specifies layers to which the logical location is connected.	0-255

1 These items are not active until you enter :AC or :AD.

Hints

:J specifies the location of the cursor relative to your block of characters. The following illustration shows the nine possible orientations.

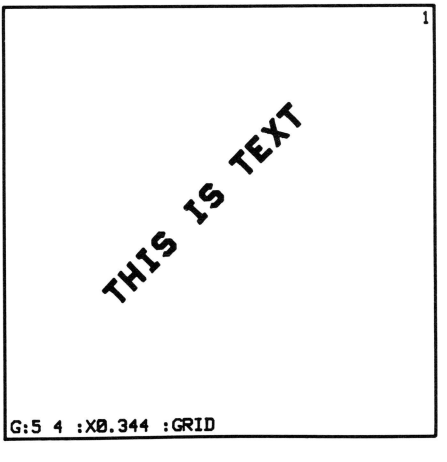


The shape of the text and note characters is defined in the EWSYS files ending with the suffix `_st,TEXT,`

Example

```
ADD T5 :F20 :FILL  "THIS IS TEXT" :R45 :J 5 -14,-10;
```

This example adds "THIS IS TEXT" on layer 5. The characters have a font size of 20, filled, and rotated 45°. The justification value of 5 centers the text about the xy point (-14,-10).

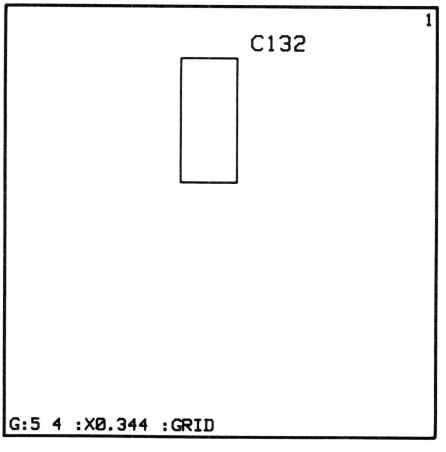


Example

```
ADD R2 -10,10 40,120;
ADD N2  :F10 :AC 40,120 "C132"  50,140;
```

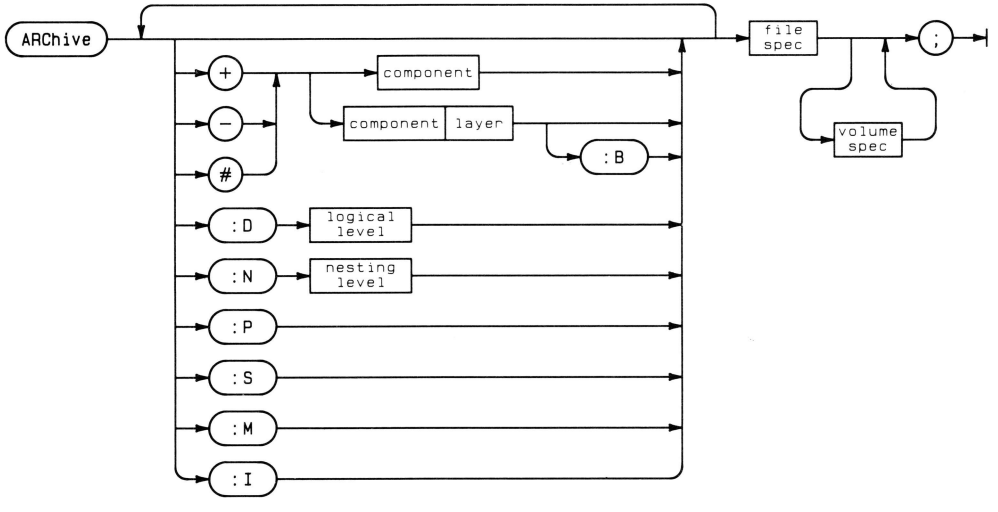
This example first adds a rectangle on layer 2 with corner points at (-10,10) and (40,120). The second command associates the note `C132` to the rectangle. The xy select point is (40,120). The note has a font size of 10 and is located at (50,140). The justification value defaults to 1 since none was specified. Therefore the xy point is located at the lower left hand corner of the boundary rectangle.

Note: Since `:AC` was specified, you could not add this note unless it could be associated to some component.



ARCHIVE

ARCHIVE creates an ASCII file that contains the Graphics Editor commands used to make a drawing. ARCHIVE does *not* smash instances and thus preserves the hierarchy of the drawing. In contrast, GENERATE smashes instances and stores the drawing instructions without preserving hierarchy. ARCHIVE is a primary command.



Item	Description/Default	Range Restrictions
Options		
+	Specifies to include in the archive file only the designated components. If a layer is also specified, only the components on that layer will be included.	
–	Specifies to include in the archive file all components except those specified by the component descriptor and layer. The default is all components on all defined layers.	
#	The same as +.	
:B	Specifies that all layers bound to the affected layer are also affected. Layers are typically bound to create logical groups.	
:D logical level	Specifies to output the archive file such that drawings less than the specified logical level are not smashed. The default is –32768.	–32768 to 32767
:N nesting level	Specifies to include instances displayed in nesting levels 1 to the specified level. For example, :N3 specifies to include only those instances in nesting levels 1, 2 and 3 of the archive file. The default is to include all nesting levels.	
:P	Specifies to output the current process file in the archive file. It is placed at the beginning of the archive file.	
:S	Specifies to smash all unnamed instances when archiving.	
:I	Specifies that implicitly stepped commands are explicitly stepped in the archive file.	
:M	Specifies that a macro instance will be archived as a graphical instance.	
Parameters		
component	One of the 13 defined descriptors.	A,C,D,E,H,I,L,M,N, O,P,R,T (See the *Components* section for descriptor names.)
layer	Layer containing the component(s) to include in or exclude from the archive file. The default is all layers included.	0-255
file spec	The drawing file you wish to archive.	8 characters maximum
volume spec	The volume in which to store the archive file.	

Hints

If you do not specify the :S option, the system names all unnamed instances and then begins the archive. The assigned names always begin with "NONAM" and end with an additional three digit name (such as NONAMA1B). When the archive file is retrieved, the system automatically removes these temporarily assigned names. It is recommended that you do *not* specify :S since unsmashed instances require less memory while archived and during retrieval.

For additional details, refer to the Archive Reference.

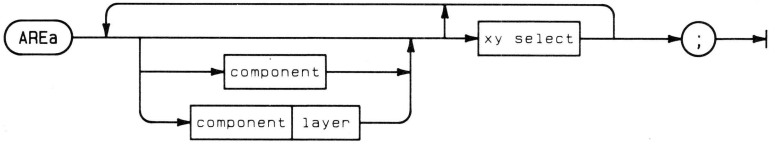
Example

```
ARCHIVE -E +L2 +A +I  :N2 EWTEMP:part1 #4: ;
```

This example first specifies to exclude all components in the archive file. However, `+L2` specifies to archive all lines on layer 2, `+A` specifies to archive all arcs, and `+I` specifies to archive instances on the drawing `part1`. `:N2` limits archived instances to nesting levels 1 and 2. The archive file is called `part1_a` and is placed on the local volume #4:.

AREA

The AREA command calculates and displays the area occupied by the component you select. If you select more than one component while the AREA command is active, a cumulative area for all components you have selected will also be displayed. Area is a secondary command.



Item	Description/Default	Range Restrictions
Parameters		
component	One of the 13 defined descriptors. The area is given for the component represented by the specified descriptor.	A,C,D,E,H,I,L,M,N, O,P,R,T (See the *Components* section for descriptor names)
layer	Layer containing the component whose area you wish to know.	1-255
xy select	xy point lying in or on the component of which you wish to know the area.	

Hints

The area display is erased when you re-enter AREA or redraw the screen.

If a component has width, only the width will be used for the area calculation.

NOTE

While in isometric mode, AREA does not calculate the true area of components.

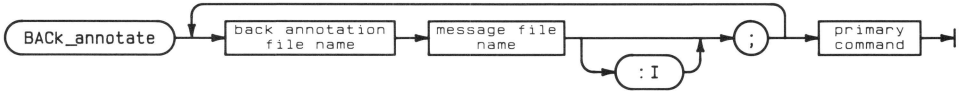
Example

```
AREA   P3 110,25  ;
```

This example calculates the area of a polygon located at (110,25) on layer 3.

BACK_ANNOTATE

The BACK_ANNOTATE command modifies the port names of a schematic drawing to match the port names of the corresponding printed circuit board. Using the back annotation file created by the Compare Connection Lists program and the program's message file, this command changes the port names either interactively or automatically for a schematic drawing currently in memory. This command does not alter port locations or modify connection lines. BACK_ANNOTATE is a primary command.



Item	Description/Default	Range Restrictions
Option		
:I	Specifies to prompt you interactively for each port name change. The system windows in on the part in question, highlights the part's reference designator and port name, and asks if you want to change the port name. When :I is not specified, the command automatically performs all port name changes.	
Parameters		
back annotation file name	The name of the file containing the back annotation information. This file is created by the Compare Connection Lists program. The command first looks for the file name as typed. If it cannot find a file with this name, it then appends an `_b` suffix and searches for this file name. If no volume name is specified, the current volume search table is used to locate the file.	
message file name	The name of the file containing information on how to interpret the back annotation file. This message file is the one used to run the Compare Connection Lists program (usually EWRC:ccomp_ms.ASC). The command first looks for the file name as typed. If it cannot find a file with this name, it then appends an `_ms.ASC` suffix and searches for this file name. If no volume is specified, the current volume search table is used to locate the file.	

Back Annotation Information

When creating a printed circuit board following the information in an electrical schematic drawing, you may need to modify the design. Thus the connections on the printed circuit board may not exactly match the connections diagrammed in the schematic. You can use the HP EGS Connection Lister and Compare Connection Lists programs to generate a list of these differences. Then you can update the schematic drawing to reflect these changes on the physical board using the BACK_ANNOTATE command.

The back annotation file produced by the Compare Connection Lists program consists of pin and gate swap information. All pertinent information lies within two delimiters, called block delimiters. The block delimiter specified in the message file is $ (the dollar sign).

In the swap information block, each line contains a port from the schematic followed by the swapped port from the printed circuit board. The format of the lines is:

```
$
part name1|ref. des.1|port name1|    part name1|ref. des.1|port name1|
part name1|ref. des.1|port name2|    part name1|ref. des.1|port name2|
part name2|ref. des.2|port name1|    part name2|ref. des.2|port name1|
$
```

The part name, reference designator, and port name information for each swap appears all on one line, with the schematic information on the left and the printed circuit board information on the right. The (vertical bar) delimiter separates the fields from one another.

For example, a back annotation file containing the line:

```
NAND2|U1A|6|     DIP14|U1|5|
```

would mean that the system should change the port named 6 on the part NAND2 with the reference designator U1A to a port named 5.

NOTE

The BACK_ANNOTATE command expects that reference designators and port names are text strings 50 characters or less. Text strings longer than this are ignored.

When the command is performing a gate swap, it changes port names for connected ports but does not change reference designators or port names for unconnected ports. This ensures that the logic of the schematic is correct, but the documentation for the schematic may not be what you intend it to be. If you want to follow a convention whereby reference designators are associated with certain sets of pin numbers (for example, U1A always contains pins 1, 2, and 3), you must change the reference designators and port names for unconnected ports in the schematic manually. To do this, use the MODIFY command to change the HP EGS notes associated to the instances of the parts. You can find a complete list of gate swaps performed by the BACK_ANNOTATE command in the report file generated by the Compare Connection Lists program.

Example

```
BACK  EWDRAW:demo1  EWRC:ccomp  :I  ;
```

This example interactively back-annotates the schematic drawing currently in memory, using the back annotation file `EWDRAW:demo1_b` and the message file `EWRC:ccomp_ms.ASC` . The system windows in on the schematic part listed in the first line of the back annotation file, highlights the part's reference designator and the port to be swapped, and asks for the port name change with a prompt like this:

```
Change port named '1' to become '2' ? (Y/N)
```

When you respond with either Y or N, the system prompts you for the next port name change, until there are no more name changes.

The changes made by the command are saved when you save the modified drawing (preferably under a new name).

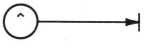
NOTE

To stop the command as it is interactively performing name changes, press the (Stop) key while a prompt is displayed . Then respond N for No to the prompt.

---

^ (backup)

Backup deletes data points one by one in the reverse order of entry. It is only effective on the following commands: GROUP, GATHER, WRAP, ADD LINE, ADD POLYGON, ADD HATCH, COPY, and WINDOW. You must enter backup before completing any of the above commands. Backup is a secondary command.



Example

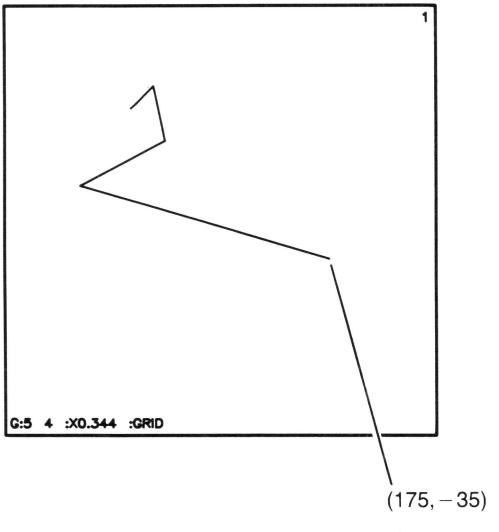
```
ADD P 0,0 20,20 30,-30 -45,-70 175,-35
```

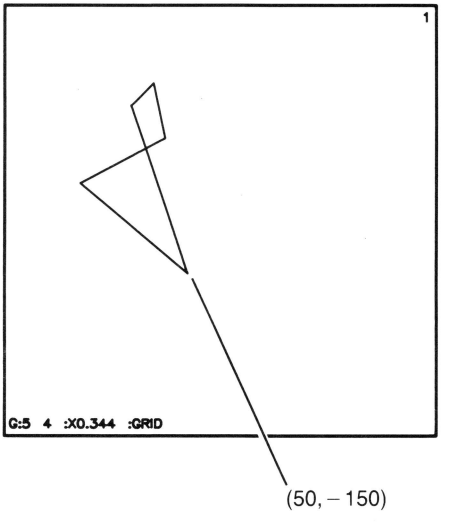
This example adds five vertices of a polygon. The polygon, which is not yet completed, appears on the left below.

The last point can be replaced with (50,-150) by entering:

```
^ 50,-150;
```

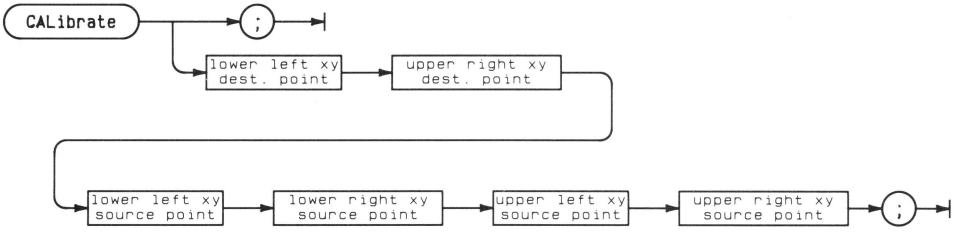
The completed polygon appears on the right below.





CALIBRATE

CALIBRATE digitizes a drawing from a graphic input device (such as a tablet) to your screen. CALIBRATE is a secondary command.



Item	Description/Default	Range Restrictions
Parameters		
lower left/upper right xy dest. point	Represents the lower left and upper right points that the digitized drawing will occupy on the screen.	
lower left xy source point	Location of the lower left corner of the drawing. You select this with the stylus. This point digitizes to the lower left xy dest. point on the screen.	
lower right xy source point	Location of the lower right corner of the drawing. You select this with the stylus.	
upper left xy source point	Location of the upper left corner of the drawing. You select this with the stylus.	
upper right xy source point	Location of the upper right corner of the drawing. You select this with the stylus. This point digitizes to the upper right xy dest. point on the screen.	

Hints

`CAL ;` serves as a toggle. If CALIBRATE mode is currently off, you can enter `CAL ;` with no parameters to reactivate the values associated with the last complete CALIBRATE command. In the same way, if CALIBRATE mode is currently on, you can enter `CAL ;` to disable CALIBRATE.

CALIBRATE can also be disabled by changing one of the current window parameters (such as `WIN :F` or `WIN :ZZ`).

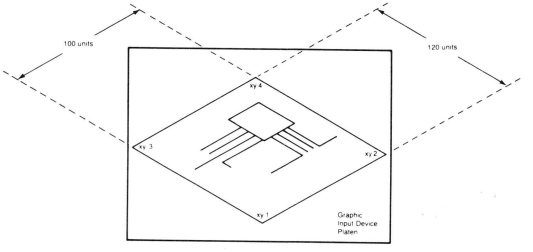
CALIBRATE will turn off the tablet menu if it is currently active.

Example

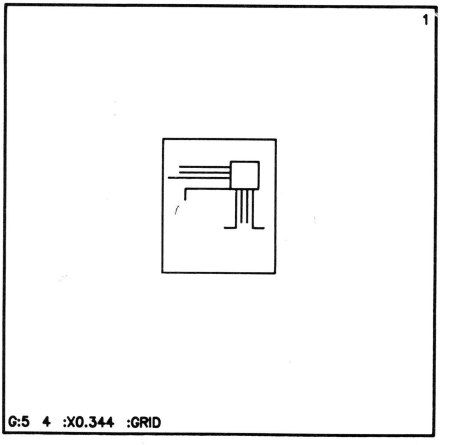
```
CAL -50,-60 50,60
```

This example begins the process of digitizing a 100 user unit by 120 user unit drawing from the tablet to the screen.

The lower left xy destination point (-50,-60) and upper right xy destination point (50,60) center the drawing on the screen area. The drawing scale is 1:1 because the digitized drawing is 100 by 120 user units on the screen.



Drawing Taped onto Tablet Menu



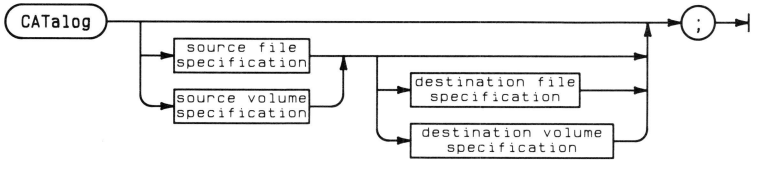
Drawing on Screen after Calibration is Complete

To continue the digitizing process:

1. Tape the drawing onto the tablet. This is because: 1) once the drawing corners are digitized, you can only access the tablet within the the area defined by the four xy source points and 2) drawing-to-screen point correspondence will be incorrect if the drawing is moved. The drawing does not have to be horizontal as the system rotates the digitized corners to fit in the screen area.
2. Select the drawing corners with the stylus. Corner selection must be in the following order: lower left, lower right, upper left, and finally upper right. NOTE: The system will always digitize the drawing as a rectangle regardless of its actual shape.
3. Check the window location parameters in the lower left corner of your screen once you have completed the calibration. The abbreviation `CAL:` should appear before the user grid setting.

CATALOG

CATALOG lists directory information about a volume or Shared Resource Manager (SRM) directory. CATALOG is a secondary command.



Item	Description/Default	Range Restrictions
Parameters		
source volume specification	Name of the volume from which to list directory information.	
destination file specification	Name of the destination file where you want to list directory information. If you do not also specify a volume, the file will be placed on the prefix volume.	
destination volume specification	PRINTER: or CONSOLE:. The default is the CONSOLE:.	

Example

```
CAT EWOPT  ;
```

This example lists the local volume EWOPT. In the illustration below, the volume name is displayed in the upper left-hand corner. To the right, the directory type is displayed. The column entries for each file include: file name, number of blocks used for storage, the file size in bytes, and the file type.

The last two lines display the number of used and unused blocks and the size of the largest space. Notice that you can tell when a disc needs to be packed by checking the unused blocks and largest space entries. A large number of unused blocks and a small largest space indicates that the unused blocks are spread over many spaces. In this example, the unused and largest block spaces are the same. The volume, therefore does not need to be packed with the PACK command. (SRM volumes do not ever need to be packed.)

```
EWOPT:              Directory type= LIF level 1
                            block size=256
Storage order

phdr,CODE              1100    281600      Code
phdr_ms,ASC             101     25856      ASCII
phdr_cdwd,TEXT            8      2048      Text
phdr_def                  1       256      Data
phdr_cn,ASC               1       256      ASCII
phdr_cnx,ASC              1       256      ASCII
run_phdr,TEXT             8      2048      Text
FILES shown=7 allocated=7
BLOCKS (256 bytes) used=1220 unused =15913 largest space=15913
```

Example

```
CAT   ;
```

This example lists the prefix volume. CAT : ; is an invalid command.

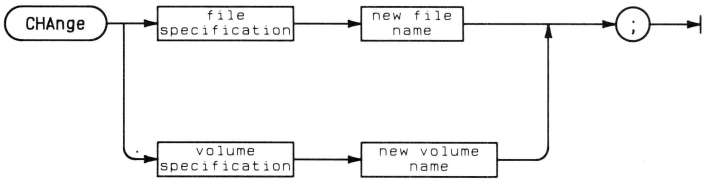
Example

```
CAT #5: PRINTER:   ;
```

This example lists the directory to the printer. CAT #6: ; and CAT printer: ; are invalid commands.

CHANGE

CHANGE renames files, directories, and volumes. CHANGE is a secondary command.



Item	Description/Default	Range Restrictions
Parameters		
file specification	Complete path specification of the file to change. You must also include the suffix of the file if it has one.	
new file name	New name for the file.	
volume specification	Name of the volume or SRM directory you wish to change.	
new volume name	New name for the volume or SRM directory.	

Hints

HP EGS uses the following file suffixes:

`_pr.TEXT`	Graphics Editor process file
`_mc.TEXT`	Graphics Editor macro file
`_mn.TEXT`	Graphics Editor screen menu file
`_tm.TEXT`	Graphics Editor tablet menu file
`_st.TEXT`	Graphics Editor stroke table
`_cn.ASC`	Control file that defines menus for the system manager
`_ms.ASC`	Message file
`_mi.ASC`	Index to a message file (for quick loading)
`_vs.TEXT`	Volume search table
`strt.TEXT`	Start file for a module
`_d`	Drawing file
`_i`	Macro instance
`_c`	Connection list output (from Connection Lister program)
`_e`	List of errors from the connection listing
`_r`	Rat's nest output (from Generate Rat's Nest program)
`_b`	Back annotation file (from Compare Connection Lists program)
`_m`	Material list output (from LIST MAT command)
`_a`	ARCHIVE output
`_g`	GENERATE output
`_ec.TEXT`	ECHO file output
`_w`	EDIF Translator warning message file

Example

```
CHA  #5:/USERS/AL/fun_d  funny_d  ;
```

This example renames the file `fun_d` to `funny_d`. The file is on the SRM directory `#5:/USERS/AL`. Note that the command includes the drawing suffix.

Example

```
CHA  EWBOB:  EWBEAR:  ;
```

This example renames the local disc `EWBOB:` to `EWBEAR:`.

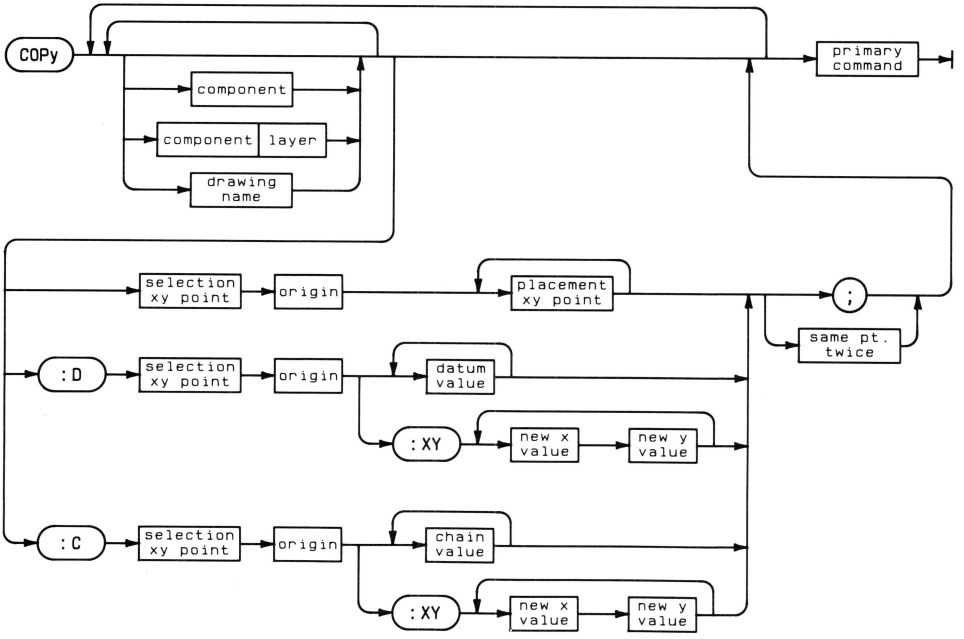
Example

```
PRE  EWDRAW  ;
CHA  oldfile_d  newfile_d  ;
```

This example shows that, if you are prefixed to the correct volume, you need only type in the CHANGE command followed by the names of the old file and the new file. You do not have to specify a volume name.

COPY

COPY copies any component in the drawing area to a new location. Multiple copies may be made by specifying additional placement points. COPY is a primary command.



Item	Description/Default	Range Restrictions
Options		
:D	Specifies that the component will be relocated a perpendicular distance from its original point. The distance (datum value) is measured from the origin to the new location. When using :D, the origin point must be on the edge of a component. When you select the origin, the cursor must be smaller than the length of the edge containing the origin.	
:C	Specifies that all copies will be placed an incremental distance from the previous component copied (chain copying).	
:XY	Specifies to copy the component the given ΔX and ΔY from its origin.	
Parameters		
component	One of 13 defined descriptors. The default descriptor is E (Every component).	A,C,D,E,H,I,L,M,N, O,P,R,T (See the *Components* section for descriptor names.)
layer	Layer occupied by the component you want to copy.	1-255
drawing name	Specifies instance to copy. This instance must be on your drawing .	8 characters maximum
selection xy point	xy location of the component to be copied.	
origin	xy location which acts as a reference point for copying.	
placement xy point	xy location where you want to place the copy of the component.	
datum value	The perpendicular distance from the original component to the copy.	
xy point	For :D this is the ΔX and ΔY distance from the original component to the copy. For :C, this is the ΔX and ΔY distance from the previously copied component.	
chain value	The perpendicular distance from the original component to the copy.	

Example

```
COPY L :C 100,0 110,0 10,20,30;
```

This example copies a line which passes through (100,0). With (110,0) as an origin, three copies will be drawn a positive distance of 10, 20, and 30 units from the original line.

Example

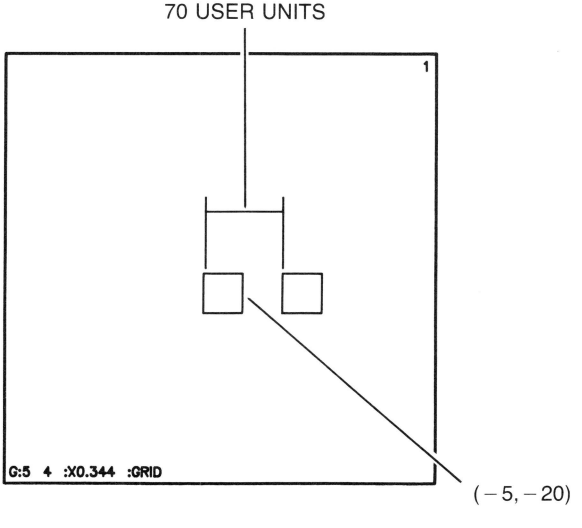
```
COP part1 -15,-45 -15,-45 35,72;
```

This example copies the instance `part1` from (-15,-45) to (35,72).

Example

```
ADD R -40,-40 -5,-5;
COP :D -5,-20 -5,-20 70 ;
```

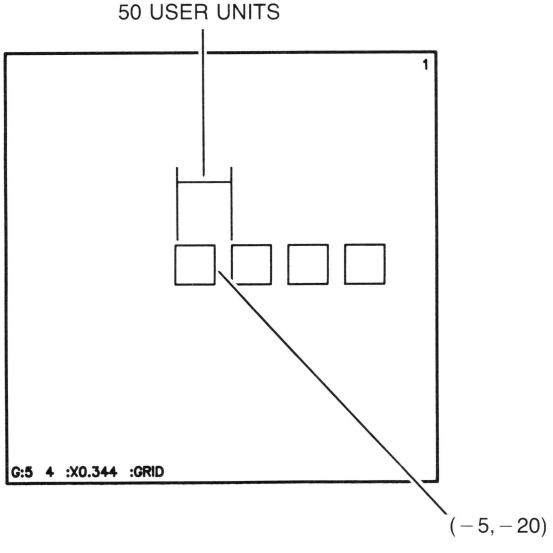
This example first adds a rectangle. The second command uses the datum `:D` option to copy the rectangle. The selection xy point and the origin are both (-5,-20). The rectangle is copied so that the origin is placed 70 user units to the right. If the datum value was -70, the rectangle would be copied to the left.



Example

```
ADD R -40,-40 -5,-5;
COP :C -5,-20 -5,-20 50 50 50;
```

This example uses the chain copy :C option to copy the rectangle three times. Each rectangle is 50 user units from the previous rectangle.



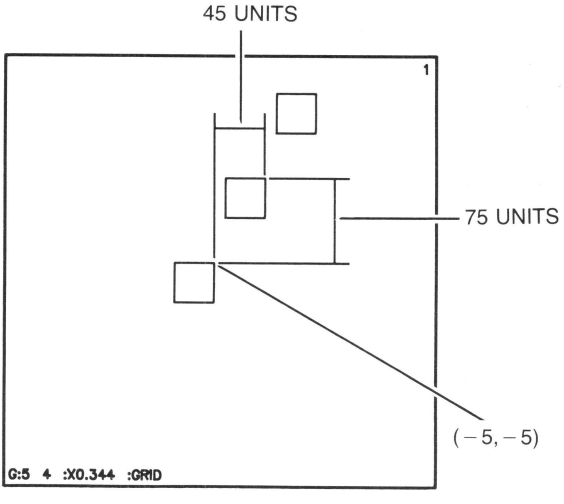
You could use :D to get the same copies by entering:

```
COP :D -5,-20 -5,-20 50 100 150;
```

Example

```
ADD R -40,-40 -5,-5;
COP :C -5,-5  -5,-5  :XY 45,75 45,75;
```

This example uses :C to make two copies of the rectangle. Since :XY was specified, each copy is 45 units in the X direction and 75 units in the Y direction from the previous copy.



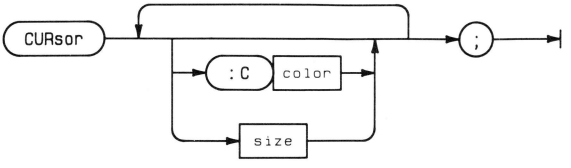
You could use :D to get the same copies by entering:

```
COP :D -5,-5 -5,-5 :XY 45,75 90,150;
```

NOTE: If the reference point is a vertex or part of a non-linear component you must use the :XY option along with the ΔX and ΔY distances.

CURSOR

CURSOR toggles between the system cursor and a user-defined cursor, defines the color of either the system or user-defined cursors, and sets the size of the user-defined cursor. CURSOR is a secondary command.



Item	Description/Default	Range Restrictions
Parameters		
:C color	Specifies the color of the cursor. The first seven colors may be specified by either a letter or a number. Colors 8 through 15 are specified by a number. W White 1 R Red 2 Y Yellow 3 G Green 4 C Cyan 5 B Blue 6 M Magenta 7 Black 8 Olive Green 9 Aqua 10 Royal Blue 11 Maroon 12 Brick Red 13 Orange 14 Brown 15 The default color is green on color CRTs and white on monochrome CRTs.	
size	Size of the user-defined cursor in user units. The system cursor is six pixels wide and the user-defined cursor defaults to the size of the screen.	0<size<1024

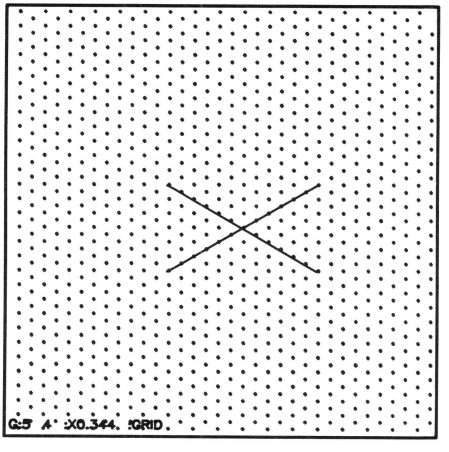
Hints

When in primitive, intersect, or vertex snapping modes, the cursor size determines the snapping area. The larger the cursor, the larger the imaginary area around the cursor in which the system searches for points when picking components. However, the system has a 40 screen dot limit when placing points rather than searching for a component to pick. If an appropriate point is not found in the window, the system snaps to a grid point.

The current grid setting determines the cursor shape. For example, if you changed the grid to isometric by entering:

```
GRI :I1 30 150;
```

the cursor would be parallel to the isometric axes as shown below.



The system cursor size never changes. Therefore, if you enter `CUR` and a size in user units, the system will toggle to the user cursor if it is not already displayed on the screen.

If you enter `CUR :C` and a color, only the cursor currently displayed on the screen changes to the new color.

Example

```
CUR    ;
```

This examples toggles from the current cursor to other cursor. For example, if the system cursor is the current cursor on the screen, then this command would toggle to the user-defined cursor.

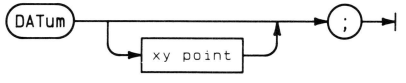
Example

```
CUR  :CR 24 ;
```

This example sets the current cursor to red and 24 user units wide.

DATUM

DATUM adjusts the position of the origin of the drawing. The new origin is indicated by a new set of crosshairs and the word DATUM. DATUM is a secondary command.



Item	Description/Default	Range Restrictions
Parameter		
xy point	New zero point (origin) for the drawing.	

Hints

DATUM can be used like COPY :D. For example, suppose you want to add a 3 by 2 unit rectangle to a point on your drawing but you do not know the exact coordinates of the point. In this case, enter

```
DAT
```

and locate the point with your stylus. This becomes the new logical origin. To add a rectangle with one corner at this new origin, enter

```
ADD R 0,0 3,2;
```

The system stores the coordinates of the upper right hand corner of the rectangle as (X+3),(Y+2), where X and Y are the coordinates of the point you just located with the stylus. Thus you can work in relative coordinates while the system converts them to absolute coordinates.

Example

```
DAT 95,10 ;
```

This example moves the origin from its present location to (95,10). This point is now the logical origin. The old origin, which is still indicated by crosshairs, now has the coordinates (−95, −10).

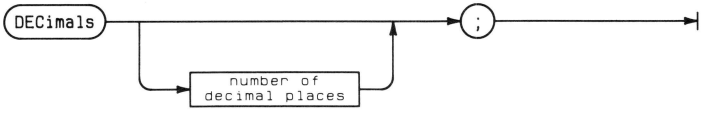
Example

```
DAT ;
```

This example resets the datum to the original origin.

DECIMALS

DECIMALS specifies the the number of decimal places for all numbers displayed on the screen. This number becomes the default value. Saving the process file saves this default value. DIM_DECIMALS controls the number of decimal places used in dimensions. DECIMALS is a secondary command.



Item	Description/Default	Range Restrictions
Parameter		
number of decimal places	Number of decimal places you want all numerical outputs (except dimensions) to have.	0-10

Example

```
DEC  3 ;
```

This example outputs all numbers to three decimal places.

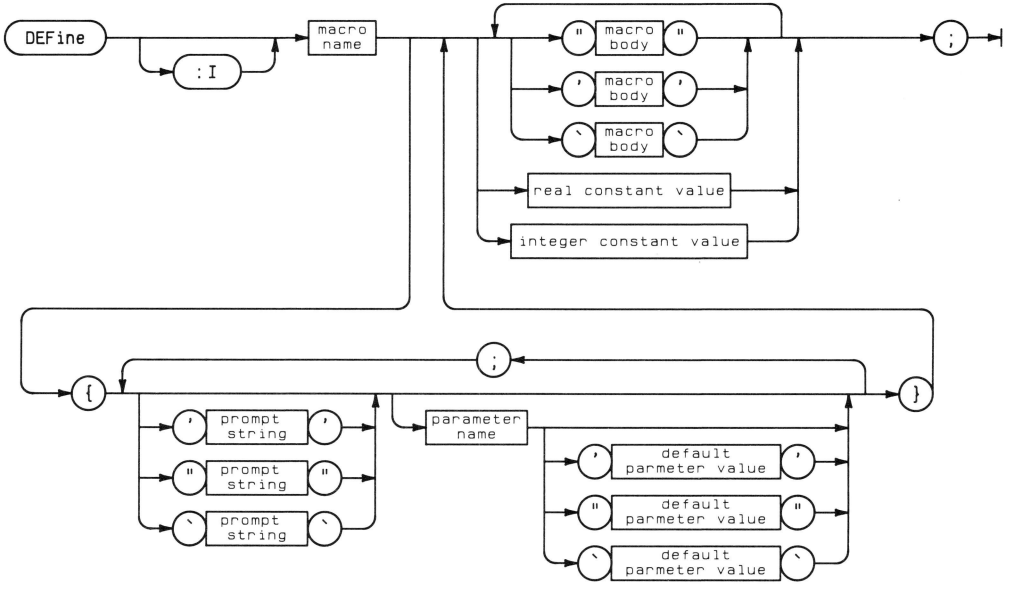
Example

```
DEC    ;
```

This example displays the number of decimal places currently being used.

DEFINE

DEFINE creates new commands, graphical instances, or macro instances for the Graphics Editor. Macros can consist of Graphics Editor commands, existing macros, or mathematical expressions. Graphical instances are drawing files and macro instances textually (in a text file) define the procedures for creating an instance. DEFINE is a secondary command.



Item	Description/Default	Range Restrictions
Option		
:I	Specifies that the macro being entered is an macro instance.	
Parameters		
macro name	Name of the command or macro instance you are defining.	20 characters maximum
macro body	A string or set of strings of commands and parameters which form the macro.	Individual strings are limited to 255 characters maximum, but you can have an unlimited number of these individual strings.
real constant value	A real number. Cannot be used with DEF :I.	
integer constant value	An integer. Cannot be used with DEF :I.	
prompt string	A user-definable prompt for any parameter in the macro.	80 characters maximum
parameter name	A placeholder for data to be entered with the newly created macro.	20 characters maximum, 12 parameters maximum
default parameter value	Default value for the parameter if no data is entered with the newly created macro. This value is kept in the recall buffer.	20 characters

Hints

You cannot enter a macro name that might be interpreted by the system as an instance, command name, library part, or layer label.

If the macro name entered is already defined, the new macro definition replaces the previous macro definition.

A macro definition may include another macro; however, the included macro must also be defined before either can be invoked. If you load a macro file, the system erases all previously defined macros.

A recursive macro calls itself and consequently puts the system into an infinite loop. To break this loop, press (STOP).

You can use the Pascal Editor to create a macro instance either by explicitly listing each command needed to create the instance or by modifying an archive or echo file. You can also graphically build an instance and then put it in a GENERATE file to produce a list of commands necessary to build the instance. Each macro instance must be contained in a separate file.

Unlike graphical instances, a macro instance must have parameter values supplied to it when added. If the macro instance is contained in an archive file, the system will use the values supplied when the instance was created as part of the original drawing. If you are adding the macro instance separately with the ADD instance command, you must supply these values. Once a macro instance has been added, it is graphically displayed and can be manipulated like any other drawing.

If a drawing which contains a macro instance is saved, the drawing will contain a reference to the macro instance file and a list of the parameters required by the macro instance.

Expressions may appear anywhere in a command or macro. The macro :

```
DEF funny 10 ;
```

defines `funny` as a constant of 10. This constant may be used as a part of other expressions.

If you use a macro frequently, you could place it on the screen menu with the MENU command. Once on the menu, the macro could be executed with the graphics tablet stylus.

For additional information on macros, see the *Customizing Your System* chapter of *Understanding HP EGS.*

Example

```
DEF WID_LINE {"Enter the layer number and line endpoints," L;x1;y1;x2;y2}
  "ADD L{L} :W10  {x1},{y1} {x2},{y2} ;";
```

This example creates a macro named WID_LINE. You could invoke this macro by entering:

```
WID_LINE 2   0,0  100,100;
```

The macro adds a 10-unit-wide line between the points (0,0) and (100,100). You could have added the same line by entering the system-level command:

```
ADD L2 :W10  0,0  100,100;
```

The macro WID_LINE has a default width of 10. If you wanted width to be a parameter, you would have to assign it a parameter name.

Example

```
DEFINE LINE  " L";
```

This creates a macro named LINE that may be used to replace the line descriptor L. NOTE: A space must precede the descriptor if it is used with the ADD command. Otherwise the system would not be able to distinguish ADDL from ADD L.

Example

```
DEF FIT  "WIN :F ;";
```

This example creates the macro FIT to execute WINDOW :FIT.

Example

```
DEFINE EXAMPLE {A;B;C}  "ADD T{A} :F{B} :R{C} 'TEXT' 0,0;" "NEWEXAMP {A},{B}"
DEFINE NEWEXAMP {X;Y} "ADD N{X} :F{Y} 'NOTE' ;"
```

This example uses one macro within the definition of another macro. Entering EXAMPLE 1, 20, 90 causes the text component TEXT to be added to layer 1 (parameter name A) . The text is 20 user units tall (parameter name B) and is rotated at an angle of 90 degrees (parameter name C). The text is placed at (0,0).

This example also shows how values entered with one macro are passed to another macro. After the text is added to your drawing, the system executes the next command in the macro body, which is the macro NEWEXAMP. The values assigned to A and B are passed to the macro NEWEXAMP and are assigned as the values of X and Y in NEWEXAMP. This causes the note component NOTE to be added to layer 1 of your drawing with a font size of 20. You must place the note by entering an xy point or selecting it with the stylus. If you wanted to rotate the note you must enter :R and a rotation value.

Example

```
DEF :I BOLTSD {"ENTer the bolt diameter," BDIA;
               "ENTer the bolt length," BLEN;
               "ENTer the bolt thread length," BTHR}
`DEF OLDSNAPMODE ("'"+SYSTEM_SNAP_MODE+"'"); :RAW `
"ADD A11 (-0.6*{BDIA}),(0.97*{BDIA}) (-0.6*{BDIA}),(0.485*{BDIA})"
       " (-0.75*{BDIA}),(0.7275*{BDIA}) ;"
       " (-0.6*{BDIA}),(0.485*{BDIA}) (-0.6*{BDIA}),(-0.485*{BDIA})"
       " (-0.75*{BDIA}),0 ;"
       " (-0.6*{BDIA}),(-0.485*{BDIA}) (-0.6*{BDIA}),(-0.97*{BDIA})"
       " (-0.75*{BDIA}),(-0.7275*{BDIA}) ;"
   " L11 (-0.6*{BDIA}),(-0.97*{BDIA}) 0,(-0.97*{BDIA})"
        " 0,(0.97*{BDIA}) (-0.6*{BDIA}),(0.97*{BDIA}) ;"
       " (-0.75*{BDIA}),(-0.7275*{BDIA}) (-0.75*{BDIA}),(0.7275*{BDIA}) ;"
       " 0,(-0.485*{BDIA}) (-0.6*{BDIA}),(-0.485*{BDIA}) ;"
       " 0,(0.485*{BDIA}) (-0.6*{BDIA}),(0.485*{BDIA}) ;"
       " 0,(0.5*{BDIA}) ({BLEN}),(0.5*{BDIA}) "
       " ({BLEN}),(-0.5*{BDIA}) 0,(-0.5*{BDIA}) ;"
       " ({BLEN}),(0.5*{BDIA}), ({BLEN}+0.15*{BDIA}),(0.35*{BDIA})"
       " ({BLEN}+0.15*{BDIA}),(-0.35*{BDIA}) ({BLEN}),(-0.5*{BDIA}) ;"
   " L14 ({BLEN}-{BTHR}),(0.5*{BDIA}) ({BLEN}-{BTHR}+0.15*{BDIA}),"
       " (0.35*{BDIA}) ({BLEN}+0.15*{BDIA}),(0.35*{BDIA}) ;"
       " ({BLEN}-{BTHR}),(-0.5*{BDIA}) ({BLEN}-{BTHR}+0.15*{BDIA}),"
       " (-0.35*{BDIA}) ({BLEN}+0.15*{BDIA}),(-0.35*{BDIA}) ;"
" OLDSNAPMODE " ;
```

This macro builds a macro instance called BOLTSD. Note that each ADD is specified in a separate string to keep the string lengths less than 255 characters. Expressions can be broken when strings are carried from one line to another if they are broken at an operator or a delimiter.

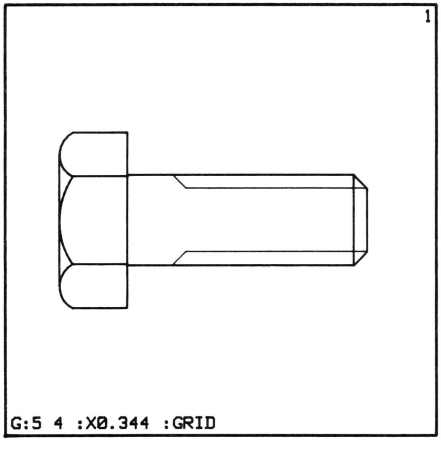
You invoke BOLTSD by entering `ADD boltsd`. Because it is a macro instance, you supply the parameter values which must be separated by a space or comma. Screen prompts ask for these values by requesting the diameter (BDIA), length (BLEN), and thread length (BTHR). The system then defines each ADD command in terms of these three parameters.

You can locate the bolt instance on the drawing by entering the xy point as part of the ADD instance command or by selecting the point with the stylus. However, if you specify a location as part of the ADD instance command, you must specify the location *after* entering the three parameter values.

Once added, you can see the bolt by windowing to nesting level 2.

The bolt can be added by entering:

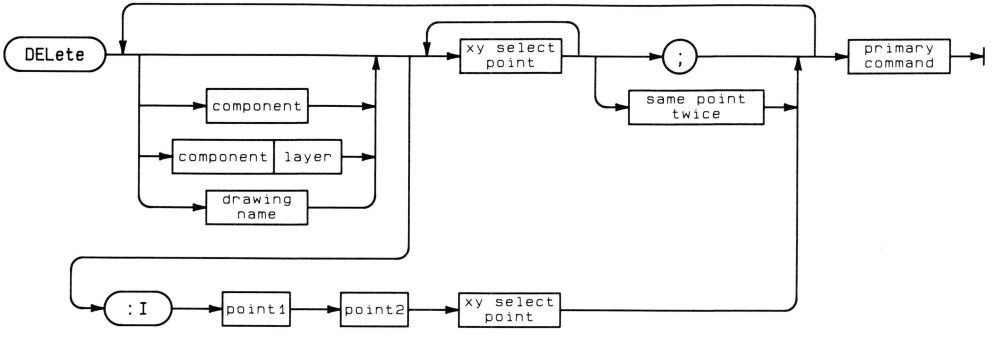
```
ADD boltsd 80 200 160 -70,0;
WIN :N2;
```



The bolt has a diameter of 80 user units, a length of 200, and a thread length of 160. The instance was located at (-70,0).

DELETE

DELETE removes any component or portion of a component from a drawing. DELETE is a primary command.



Item	Description/Default	Range Restrictions
Option		
:I	Specifies to delete a portion of a component. Once a component has been altered in this way, it is no longer considered by the system to be part of any collection (see GATHER).	
Parameters		
component	One of 13 defined descriptors.	A,C,D,E,H,I,L, M,N,O,P,R,T (See *Components* section for descriptor names.)
layer	layer on which you are deleting the component.	1-255
drawing name	Name of drawing file to be deleted.	
xy select point	xy location on the component to be deleted.	
point 1	First point of the partial component to be deleted.	
point 2	Second point of the partial component to be deleted.	

Example

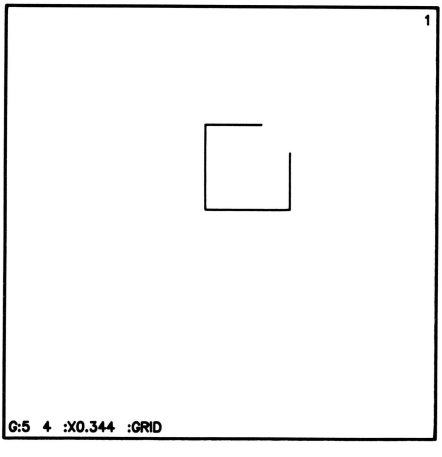
```
DEL L2 100,200 ;
```

This example deletes the last line component which was added on layer 2 and passes through (100,200).

Example

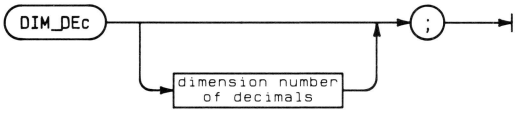
```
ADD R 125,125 50,50 ;
DEL :I :PRI 100,125 125,100 125,125 ;
```

This example deletes the upper right corner of the rectangle. Primitive snapping mode was used since the deleted portion was from a single component.



DIM_DEC

DIM_DEC changes the number of decimal places of dimension outputs. This becomes the default value. Saving the process file saves this default. DIM_DEC is a secondary command.



Item	Description/Default	Range Restrictions
Parameters		
number of decimal places	Number of decimal places for dimensions outputs.	0-10

Example

```
DIM_D   3 ;
```

This example sets the dimension output to three decimal places.

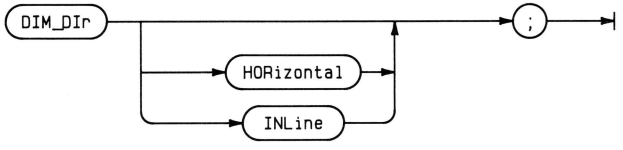
Example

```
DIM_D   ;
```

This example outputs the current number of decimal places for dimensions.

DIM_DIR

DIM_DIR changes the orientation of dimension values relative to dimension lines. This becomes the default value. Saving the process file saves this default. DIM_DIR is a secondary command.



Item	Description/Default	Range Restrictions
Options		
HORizontal	Specifies to display dimension values horizontally regardless of the orientation of the dimension lines. This is the default.	
INLine	Specifies to display dimension values in line with the dimension lines. :L dimension vales will be horizontal if the dimension line is vertical.	

Example

```
DIM_DI  HOR ;
```

This example specifies that dimension values will be horizontal.

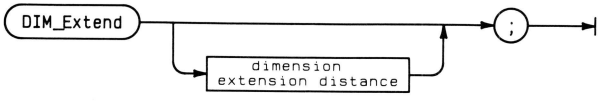
Example

```
DIM_DI  ;
```

This example displays the current orientation of dimension values.

DIM_EXTEND

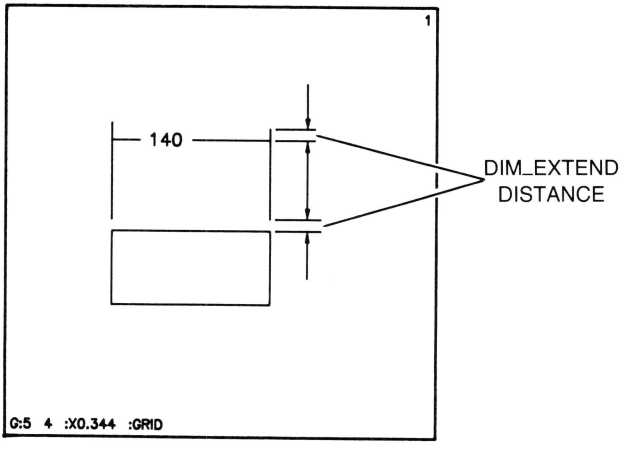
DIM_EXTEND changes the distance between dimension extension lines and the object being dimensioned. The extension lines also extend past the dimension line by the same distance. DIM_EXTEND changes this distance as well. This distance becomes the default value. Saving the process file saves this default. DIM_EXTEND is a secondary command.



Item	Description/Default	Range Restrictions
Parameter		
dimension extension distance	Default distance for the dimension extension.	0 to 10^9 divided by internal units per user unit.

Hints

The DIM_EXTEND distances are illustrated below:



Example

```
DIM_E   0.125 ;
```

This example sets the width between the dimensioned object and the extension line to 0.125 user units.

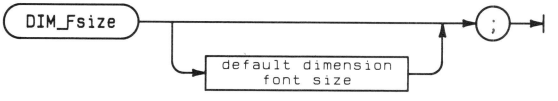
Example

```
DIM_E  ;
```

This example displays the current dimension extension distance .

DIM_FSIZE

DIM_FSIZE changes the font size of dimension labels. This becomes the default value. Saving the process file saves this default value. DIM_FSIZE is a secondary command.



Item	Description/Default	Range Restrictions
Parameter		
text font size	Dimension font size in user units.	>0

Example

```
DIM_F  0.25  ;
```

This example sets the dimension font size to 0.25 user units.

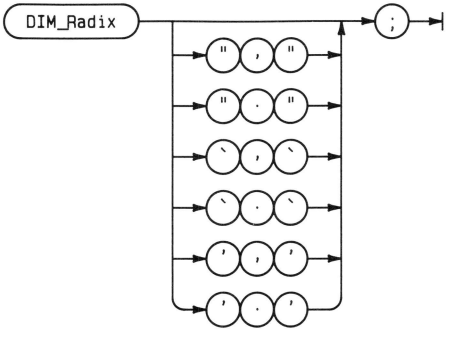
Example

```
DIM_F ;
```

This example displays the current font size for dimensions.

DIM_RADIX

DIM_RADIX changes the radix (delimiter) of dimension values. This becomes the default value. The radix can be either a period or comma. (European convention separates integers from decimals with a comma rather than a period; for example, 10,000 instead of 10.000.) Saving the process file saves the DIM_RADIX default value. DIM_RADIX is a secondary command.



Item	Description/Default	Range Restrictions
Parameters		
	Specifies that the radix is a comma (,).	
	Specifies that the radix is a period (.).	

Example

```
DIM_R "," ;
```

This example sets the radix to a comma.

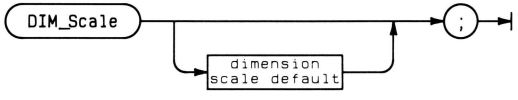
Example

```
DIM_R  ;
```

This example displays the current radix.

DIM_SCALE

DIM_SCALE changes the scale value of dimensions. This scale becomes the default value. Saving the process file saves the DIM_SCALE value. DIM_SCALE is a secondary command.



Item	Description/Default	Range Restrictions
Parameter		
dimension scale default	Specifies to divide the calculated dimension value by the given default value. The process file default is 1.	

Use of DIM_SCALE

The DIM_SCALE value should be changed before dimensioning a component (such as a library part) which has a scale different than one. In this case, change the DIM_SCALE value to the scale of the component. For example, suppose you add a hub section to your drawing. The hub has a six inch diameter and to fit it into the available drawing space, you scale it to half size. At this scale the hub has a three inch diameter. If you dimensioned the hub at this time, the dimension value would be 3. However, to correctly dimension the scaled down hub, first enter `DIM_SCALE .5;`. Then when you dimension the hub, the dimension value would be 6.

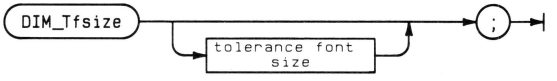
Example

DIM_S;

This example displays the current scale value of dimensions.

DIM_TFSIZE

DIM_TFSIZE changes the font size of the dimension tolerance values. This becomes the default. Saving the process file saves this default value. DIM_TFSIZE is a secondary command.



Item	Description/Default	Range Restrictions
Parameter		
tolerance font size	Specifies the tolerance font size in user units.	>0

Example

```
DIM_T    0.125 ;
```

This example changes the tolerance font size to 0.125 user units.

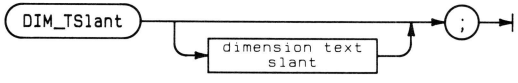
Example

```
DIM_T ;
```

This example displays the current tolerance font size.

DIM_TSLANT

DIM_TSLANT changes the text slant of dimension values. This becomes the default slant. Saving the process file saves this default slant. DIM_TSLANT is a secondary command.



Item	Description/Default	Range Restrictions
Parameter		
dimension text slant	Specifies the angle of slant of the dimension text. The default is 0°.	-75° through 75°

Example

```
DIM_TS      30 ;
```

This example changes the slant of dimension text to 30° to the right.

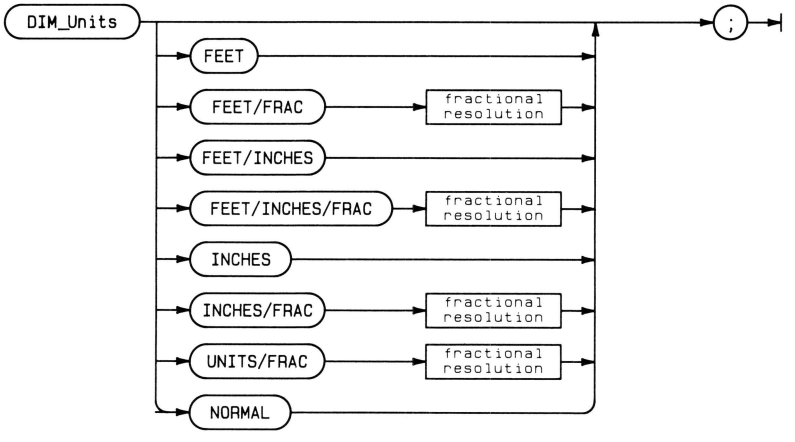
Example

```
DIM_TS ;
```

This example displays the current text slant of dimension values.

DIM_UNITS

DIM_UNITS changes the units of line, radius, and diameter dimensions. These units become the default value. Saving the process file saves this default value. DIM_UNITS is a secondary command.



Item	Description/Default	Range Restrictions
Options		
FEET	Specifies to display dimension values in feet.	
FEET/FRAC	Specifies to display dimension values in feet and fractions of a foot.	
FEET/INCHES	Specifies to display dimension values in feet and inches.	
FEET/INCHES /FRAC	Specifies to display dimension values in feet, inches, and fractions of an inch.	
INCHES	Specifies to display dimension values in inches.	
INCHES/FRAC	Specifies to display dimension values in inches and fractions of an inch.	
UNITS/FRAC	Specifies to display dimension values in the current user units defined and fractions of those units.	
NORMAL	Specifies to display dimension values in user units with the default number of decimals.	
Parameter		
fractional resolution	Specifies how many parts there are to one unit. The system rounds up to the given fractional resolution. For example, if the fractional resolution is 4, a dimensioned object 9/64ths of a user unit long would read 1/4.	

Example

```
DIM_U     FEET/INCHES/FRAC 64 ;
```

This example changes the default dimension units to feet, inches and fractions (1/64) of an inch.

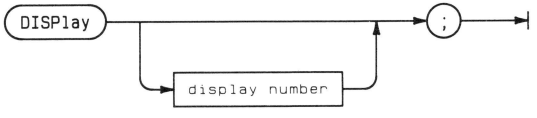
Example

```
DIM_U     ;
```

This example displays the current dimension units.

DISPLAY

DISPLAY specifies whether the video output goes to an internal display screen (CRT) or to a video interface card, and how the video interface card is to be configured. DISPLAY is a secondary command.



Item	Description/Default	Range Restrictions
Parameter		
display number	Line number in EWCODE:gedit_ms.ASC associated with the DISPLAY tag that lists the possible display devices. If you specify a display number greater than the number of devices defined in the tag, the computer displays an error message and ignores the command.	1 through the number of lines in the display tag

Describing Display Devices

Each line of the DISPLAY tag describes a single device. The line contains two numbers. The first is the select code of the display to be used when this display number is selected. The second number corresponds to the control value to be used in initializing the display.

The following list defines the values included in the standard EWCODE:gedit_ms.ASC message file.

Item	Description/Default	Range Restrictions
Display number	Message File Entry	
1	3 0	Internal CRT
2	28 256	HP 98627A I/O Card, US Standard
3	28 512	HP 98627A I/O Card, European Standard
4	28 768	HP 98627A I/O Card, US TV
5	28 1024	HP 98627A I/O Card, European TV
6	28 1280	HP 98627A I/O Card, High Resolution
7	6 0	Secondary CRT

Example

```
DISP 2      ;
```

This example specifies that the graphics are sent to a US standard monitor via an HP 98627A color card. The alpha display still appears on the internal display screen.

Example

```
DISP ;
```

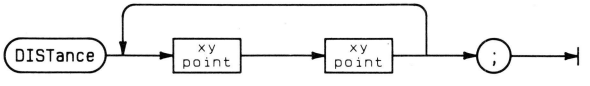
This example redraws the screen (including menus) to the currently active display device.

NOTE: HP EGS does not support Katakana menu or screen message characters. The display will, however, draw Katakana note and text components.

DISTANCE

DISTANCE displays the x, y, and vector distance (in user units) between two specified points. This information is erased when the screen is redrawn. DISTANCE is a secondary command.

Note that the system cannot measure true distance across isometric planes.



Item	Description/Default	Range Restrictions
Parameter		
xy points	Points between which the system calculates the distance.	

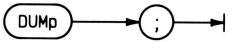
Example

```
DISTANCE 0,0  100,120 ;
```

This example calculates the x, y, and vector distances between (0,0) and (100,120).

DUMP

DUMP outputs the graphics screen to a local printer. DUMP is a secondary command.



Hints

If you have a DUMP GRAPHICS key on your keyboard, you can use it to dump graphics displays to your local printer. Similarly, if you have a DUMP ALPHA key on your keyboard, you can use it to dump alpha displays one screen at a time to your local printer.

The DUMP command will not give an image of the full screen width for all combinations of displays, printers, and paper widths.

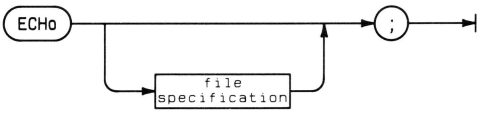
Example

```
DUMP ;
```

This example dumps the graphics screen display to your local printer.

ECHO

ECHO outputs the commands and data you have entered as well as system messages. ECHO is a secondary command.



Item	Description/Default	Range Restrictions
Parameter		
file specification	Specifies the file where the data will be sent. The system appends the suffix `_ec.TEXT` to echo files.	6 characters maximum

Example

```
ECH ;
```

This example begins to output your commands, data, and any system messages to the local printer. This command acts as a toggle. If an echo file was already in progress, `ECH ;` would toggle off the echo.

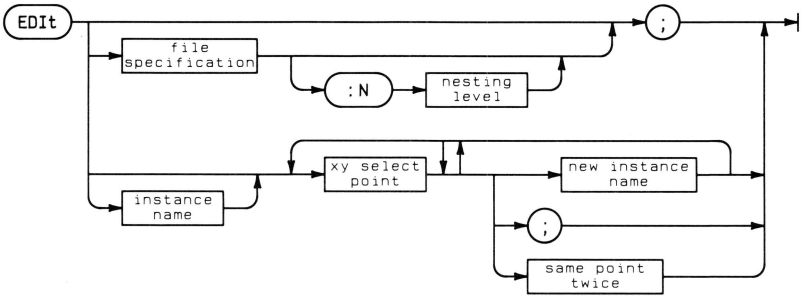
Example

```
ECH  EWSYS:temp;
```

This example lists the commands, data, and system messages entered during a drawing session to a file named temp_ec.TEXT on volume EWSYS.

EDIT

EDIT is used to create a new drawing, modify an existing drawing, or modify an instance within a drawing. You may use any Graphics Editor command to do these modifications. Entering the EDIT command without parameters (`EDI ;`) provides a clean display screen on which you can make a new drawing. EDIT is a primary command.



Item	Description/Default	Range Restrictions
Options		
:N nesting level	Indicates the nesting level at which you want the drawing loaded into memory and displayed.	1 - 32767.
Parameters		
file specification	Name of the drawing you wish to edit.	8 characters maximum.
instance name	Name of the existing instance you wish to edit. Aids in search of dense drawings.	8 characters maximum; name must be unique.
xy select point	Specifies a point which lies on an instance to be edited. The contents of the instance must be visible.	
new instance name	New name for next instance to be edited in context.	8 characters maximum; name must be unique.

Use of EDIT

The following discussion explains EDIT by its two primary functions: as a graphical EDIT and as a instance-in-context EDIT. The command path of the graphical EDIT follows the upper branch of the EDIT syntax diagram. The command path of the instance-in-context EDIT follows the lower branch of the syntax diagram.

Graphical EDIT

As illustrated by the syntax diagram, you can:

- Enter `EDI ;` to get an empty screen. If a drawing is already on the screen, the system places this drawing, if named, in memory. If the existing drawing is not named, the system prompts you to name this drawing or lose it.

- Enter `EDI` with a file specification. If a drawing is already on the screen, the system places this drawing, if named, in memory. If the existing drawing is not named, the system prompts you to name this drawing or lose it. Once the current drawing is removed, the drawing of the file specification appears on the screen. If no drawing exists with the file specification, a clean sheet appears that has the file specification name.

  When you enter `EDI` with a volume and file specification, the system will restrict its search to the specified volume. If you enter a suffix, the system only searches for the type of file represented by the suffix.

  When you enter `EDI` with a file specification but with no volume or suffix specification, the system will search current memory for a drawing or macro instance with the file specification. If it finds a drawing, this drawing is placed on the screen. If it finds a macro instance, the system switches to the text editor and displays the textual definition of this macro.

  If the system cannot find a drawing or macro instance in current memory which matches the file specification, and no suffix or volume has been given with this specification, the system will begin searching the volumes in the search table for the specified drawing file. If it cannot find a drawing file, it begins looking for macro instances. If the system cannot find any macro instances, it draws a clean sheet appears on the screen that has the file specification name.

- Enter `EDIT`, a file name, `:N` and a nesting level. The drawing is loaded only to the nesting level you have specified. The :N option is only valid if the specified file is a drawing file.

- Edit an instance within another drawing if the instance has a file specification. Any changes you make to the instance, if later saved under the same name, will also appear on the other drawing(s) in which the instance is contained.

- Edit a macro instance. When editing a macro instance, the system switches to the text editor specified by the `text_editor_name` line of the gedit_ms.ASC file. If you already have a workfile, it will appear rather than the macro instance. You can delete this workfile by loading the Filer and entering `N` for NEW. When you leave the text editor the system returns to the location, or state, in which you originally entered the EDIT command. If a drawing file has the same name as the macro instance you wish to edit, the system will call up the drawing file unless you specify the suffix `_i` after the macro instance name. The :N switch is not valid for macro instances.

NOTE:

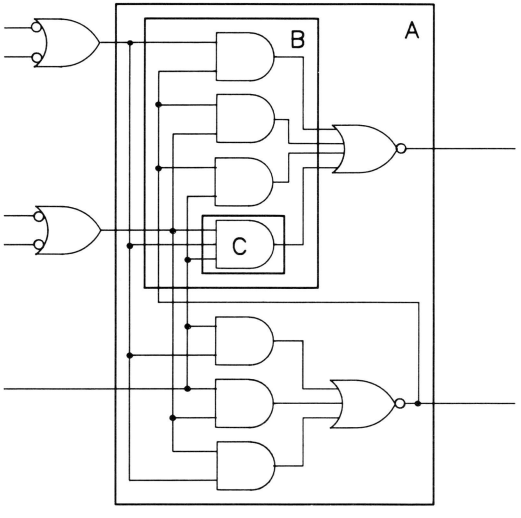
You can only perform a graphical edit while in nesting level one. For example, if you are already editing an instance in context, you must return to nesting level one to edit other parts of the drawing.

Instance-in-Context EDIT

In contrast to the global graphical EDIT, the instance-in-context EDIT focuses on a particular instance. You can specify this instance by entering `EDI`, an xy select point which lies on the instance, and optionally, the name of the instance.

The instance you wish to edit must be on a nesting level greater than or equal to the one you are currently on. However, you must be able to see this instance. For example, if you wish to edit an instance on nesting level 4, you must first window to at least nesting level 4 before entering `EDI`.

When you decide to edit an instance which lies on a nesting level deeper than the one you are currently on, you must first consider whether or not you wish to change the names of the instances in which your particular instance lies. In the illustration below, instance C is nested within B which in turn is nested in A.



If you wish to edit C, you can do one of two things to the instances A and B: 1) Make them *unnamed* instances or 2) *Rename* them.

1) *Unnamed:* Enter `EDI`, an xy select point on C, (the instance name C is optional) and a semicolon. Instance A will become an unnamed instance. From that time on, it can only be referenced on that drawing by selecting a point which lies on that instance.

2) *Renamed:* Enter `EDI`, an xy select point on C, and a new instance name. Instance A will be changed to that name. If you save the renamed A, you can later call it up to use on this or any other drawing. (The new instance name must be unique to the editing session and if you plan to save it to a disc, the name must also be unique to that disc.) Macro instances cannot be renamed.

The instance one level down from where you are to begin the edit is highlighted as an editable instance. Only an editable instance can be edited in context at this time. In this example, A is the editable instance because it is the first instance and one level down from where you began the edit.

The instance C, which the system labels the target instance, still lies two nesting levels deeper. It is also highlighted. However, it cannot yet be edited. To edit C, you must continue down level by level with the edit command. The way you continue depends on whether you changed A to an unnamed instance or renamed it.

If you changed A to an *unnamed* instance by entering `EDI`, the xy select point, and a semicolon, you must re-enter the `EDI` again before C becomes both the editable and target instance. If you *renamed* A by entering `EDI`, the xy select point, and a new instance name, the system will prompt you by asking:

```
Do you wish to continue traversing the path: (Y/N)
```

If you enter "y" for yes, the system will essentially re-enter the EDIT command and the xy select point for you . The screen prompt will now say the instance B is the editable instance and C is still the target instance. You can rename B by entering another instance name at this time or make B an unnamed instance by not entering another instance name. If you enter "n" for no, the system un-highlights the instances and allows you to edit the editable instance in its context.

Notice that after each entry of EDIT, a number is displayed within < >. This number shows the number of times you have entered a context EDIT. The number is also one less than the nesting level of the current editable instance.

Once you have completed your necessary editing of instances-in-context, you step back out with the EXIT command. Each time you enter `EXI`, you return to the instance at the previous nesting level. For example, if you had edited C and then exited, you would now be on nesting level 3 and instance B.

If the instance you have exited from has not been saved, the system will prompt:

```
This level of context edit has not been saved.  If you continue, you will
 lose all changes made at this level and below.  Do you wish to continue:
 (Y/N)
```

If you type in "y" for yes, the instance will remain unchanged under its original name. If you type in "n" for no, the system does not exit and you can save your edited instance .

If you exit any level without saving it , any saves you have done on unnamed instances on deeper levels will be lost. For example, if you save the changes made on instance C (if it has not been renamed) exit, and then exit from B without entering `SAV`, you will lose the changes on C. However, if you have renamed C and then save it, an exit without a save from a higher level will not affect the disc copy of the C instance. However, any changes in current memory will be lost.

NOTES:

1. If an instance was created as an implicit step (by `ADD :I`), the system will call up the original image of the instance for you to edit. Any changes will be reflected by all of the implictly stepped images .

2. You can modify only the parameter values of a macro instance edited-in-context by entering `EDI` and the xy point of the instance. When the parameter prompt appears on the screen, press **f2** (the **RECALL** softkey) to see the current parameter value.

   If you do not wish to modify this parameter value, press (Return). If you do wish to modify it, use the editing keys to replace the value. Then press (Return).

   You cannot modify the textual instructions of a macro instance without graphically editing it. For an example of a macro instance see the DEFINE command.

Example

```
EDIT #3:exampl ;
```

This example calls up the drawing file #3:exampl.

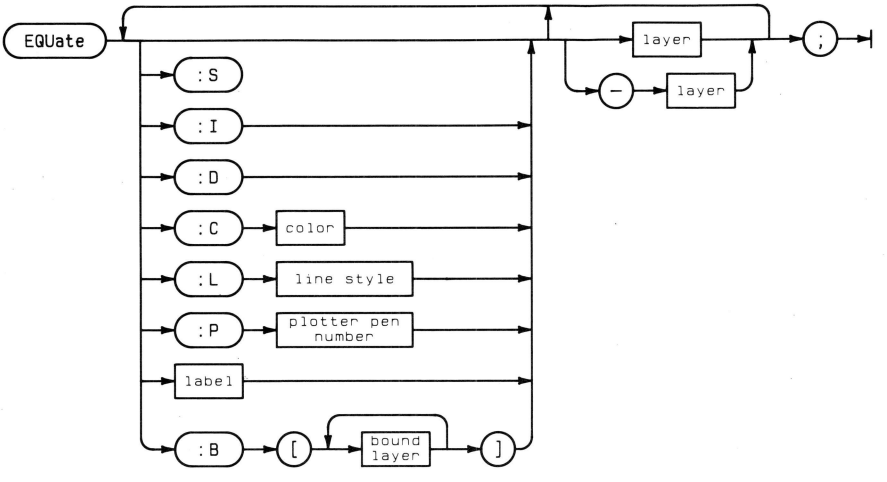
Example

```
EDIT trans  0,0  trans1 ;
```

This example specifies that the instance `trans` is to be the target instance for an edit-in-context. To be the first editable instance, `trans` must lie only one nesting level deeper than the user is currently at. If `trans` lies more than one level down, the instance which is one level down and contains `trans` becomes the first editable instance and is renamed `trans1`.

EQUATE

EQUATE defines each layer you plan to use in a drawing. To preserve the layer definitions, save the process file before leaving the Graphics Editor, or edit and save the process file. EQUATE is a secondary command.



Item	Description/Default	Range Restrictions
Options		
:S	Specifies that the layer is a symbolic layer.	
:I	Specifies that the layer is an interconnect layer.	
:D	Specifies that the layer is a detail layer. This is the default layer.	
Parameters		
layer	Number of the layer being defined.	0-255
–	Specifies to delete the current definition of the specified layer. The default attributes for that layer are assumed.	
:L line style	Specifies the line style by number. The style is ignored when :FILL is specified. The default line style is solid.	1-8

Line Type	Number	Result
solid	1	
dashed	2	
long dash	3	
dot center	4	
dash center	5	
phantom	6	
long dotted	7	
dotted	8	

(Table continued)

Item	Description/Default	Range Restrictions
Parameters (cont)		
:C color	Specifies the color of the components on the layer being defined. The first seven colors may be specified by either a letter or a number. Colors 8 through 15 are specified by a number W White 1 R Red 2 Y Yellow 3 G Green 4 C Cyan 5 B Blue 6 M Magenta 7 Black 8 Olive Green 9 Aqua 10 Royal Blue 11 Maroon 12 Brick Red 13 Orange 14 Brown 15 The default color is green on color CRTs and white on monochrome CRTs.	
:P plotter pen number	Plotter pen number with which to plot components in this layer. The default pen number is 0.	0-8
:B bound layer	Specifies which other layers are bound to the specified layer. The other commands using this information are: ARCHIVE, GENERATE, LIST MATERIAL, and SHOW.	0-255
label	A note which describes the contents of the layer. The default is an empty string.	8 characters maximum

Use of EQUATE

The EQUATE command defines layers. To define new layers, simply specify EQUATE followed by a layer number. Unless you specify other attributes, the default attributes are assumed.

Once a layer is defined, you can reset some or all of that layer's attributes to be what you want or to be the default values. It is possible to change the attributes for layer 0, the layer reserved by the system for displaying instances of library parts. However, layer 0 cannot be removed from the defined layer set.

Example

```
EQUATE :D :CY :L6 :P3  1,3,5;
```

This example defines layers 1, 3, and 5 to be detail layers. Components entered on these layers are drawn in yellow (on systems with a color CRT) and in phantom line style. When plotted, components on these layers are plotted using pen number 3.

If you then specify:

```
EQUATE :CR 1,3;
```

components on layers 1 and 3 are drawn in red. The line style stays phantom, as this was the previously defined line style. Specifying:

```
EQUATE -1,-3;
```

deletes layers 1 and 3 from the defined layer set. Specifying:

```
EQUATE 1;
```

then defines layer 1 with all of the default attributes, since this layer was previously removed from the defined layer set.

EVALUATE

EVALUATE performs simple mathematical calculations while in the Graphics Editor. The value of the result appears in both the alpha and graphics displays. EVALUATE is a secondary command.

Item	Description/Default	Range Restrictions
Parameter		
expression	Integer or real expression you wish to evaluate.	

Constants, Variables, Operators, and Functions

The expression may contain constants, variables, operators, or functions.

Constants

HP EGS system constants may be evaluated. The system constants are listed in *Understanding HP EGS*–Customizing Your System.

Variables

A variable is a name and a value created with the DEFine command . The variable name can be constructed from letters, numbers and the underscore character. It must start with a letter, be in upper or lower case , and is limited in length to 20 characters. The defined variables must be integers, real numbers, or strings of less than 250 characters.

Operators

Numeric operators

+ for addition of numeric operands and concatenation of string operands

– for negation and subtraction

* for multiplication

/ for division

^ for exponentiation

The precedence for operators from high to low is negation (–), exponentiation (^), multiplication or division (*,/), and addition or subtraction (+, –).

Logical operators

AND, OR, NOR and NOT. NOT has higher precedence than the others. A boolean expression is returned as an integer: 1 for true and 0 for false.

Operators

Relational Operators

= equal to.
<> not equal to.
< less than.
> greater than.
<= less than or equal to.
>= greater than or equal to.

Functions (must be in upper case.)

Scientific functions

SIN(x) sine of x, x in degrees

COS(x) cosine of x, x in degrees

ARCTAN(x) angle in degrees whose tangent is x

SQR(x) square of x

SQRT(x) square root of x

LN(x) natural logarithm of x

EXP(x) e to the xth power

ABS(x) absolute value of x

TRUNC(x) x truncated to an integer

ROUND(x) x rounded to an integer

IFELSE (expression, true result, false result) Returns the evaluation of one of two clauses depending on whether the expression is 0 (false) or non-zero (true).

String functions

SUBSTR(s,i,l) Return a substring of s starting at the ith character of s and continuing for l characters

LENGTH(s) Return the length of string s

NUM2STR(numeric expression) Converts the numeric expression into a string.

NUM2STR(numeric expression, #) Converts the numeric expression into a string, truncating it the specified number (#) of places to the right of the decimal point.

CHR (integer expression) Converts the ordinal value of the integer expression to the corresponding ASCII character.

STRPOS(source, pattern) Where source and pattern are string expressions, STRPOS returns the integer value equal to the sequence number in 'source' of the first character in the matching pattern.

Example

```
EVA ((((5*6)*(5+6))*SIN(45))*100);
```

This example calculates the above expression and prints 23334.3, in both the alpha and graphics areas of the display.

Example

```
EVA (NUM2STR(9.3028472,3));
```

This example truncates the expression to the string 9.303.

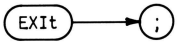
Example

```
ADD R3 (4*4),(3*4)  (SIN(45)*10),(SQR(20));
```

This example uses expressions in a command other than EVALUATE. This example adds a rectangle with corner points at (16,12) and (7.07,400).

EXIT

EXIT ends an editing session of an instance on a particular nesting level and then brings you to the next higher level. The use of EXIT is explained in the EDIT command section. EXIT is a primary command.



Hints

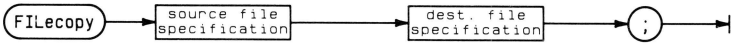
Entering an EXIT when you have not been editing an instance in context causes the command to be ignored.

UNDO will not reverse the effects of EXIT.

For additional information, see the *Instance in Context EDIT* section of the EDIT command.

FILECOPY

FILECOPY copies a specified source file to a specified destination file. FILECOPY is a secondary command.



Item	Description/Default	Range Restrictions
Parameters		
source file specifier	Name of file to be copied.	
dest. file specifier	Name of destination file.	

Hints

The HP EGS version of the Pascal Filer does not allow the use of wildcards (such as $, =, ?).

When specifying a file name, you must include the complete file path and the appropriate suffix. The suffixes used by HP EGS are:

`_pr.TEXT`	Graphics Editor process file
`_mc.TEXT`	Graphics Editor macro file
`_mn.TEXT`	Graphics Editor screen menu file
`_tm.TEXT`	Graphics Editor tablet menu file
`_st.TEXT`	Graphics Editor stroke table
`_cn.ASC`	Control file that defines menus for the system manager
`_ms.ASC`	Message file
`_mi.ASC`	Index to a message file (for quick loading)
`_vs.TEXT`	Volume search table
`strt.TEXT`	Start file for a module
`_d`	Drawing file
`_i`	Macro instance
`_c`	Connection list output (from Connection Lister program)
`_e`	List of errors from the connection listing
`_r`	Rat's nest output (from Generate Rat's Nest program)
`_b`	Back annotation file (from Compare Connection Lists program)
`_m`	Material list output (from LIST MAT command)
`_a`	ARCHIVE output
`_g`	GENERATE output
`_ec.TEXT`	ECHO file output
`_w`	EDIF Translator warning message file

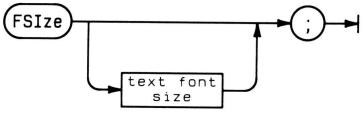
Example

```
FILECOPY  EWSYS:funny_d  EWBEAR:funny_d  ;
```

This example copies the file `funny_d` from the volume EWSYS to the volume EWBEAR. The file name is unchanged.

FSIZE

FSIZE sets the font size of notes and text. This becomes the default size. Saving the process file saves this default. FSIZE is useful in drawings which require that all printed information be of a standard height. FSIZE is a secondary command.



Item	Description/Default	Range Restrictions
Parameter		
text font size	Default font size for notes and text.	>0

Example

```
FSI 5 ;
```

This example changes the default font size to 5 user units.

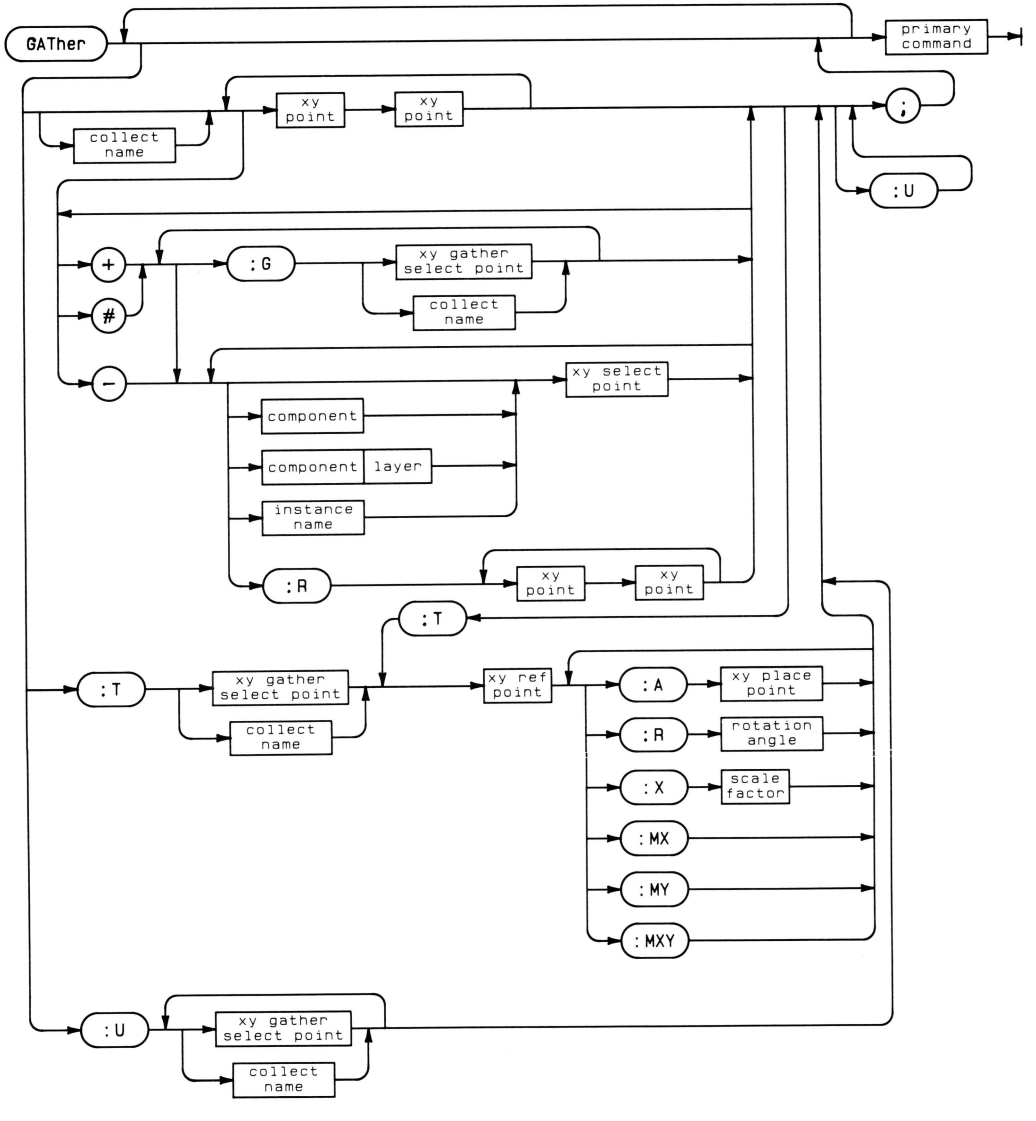
Example

```
FSI  ;
```

This example displays the current default font size.

GATHER

GATHER groups individual components and component vertices into a collection. This command is primarily used to adjust a collected group of vertices without breaking their connections. Unlike WRAP, the components and vertices of a GATHER collection are NOT displayed in the next higher nesting level. A collection is not preserved when a drawing is saved. GATHER is a primary command.



Item	Description/Default	Range Restrictions
Options		
+	Include the vertex or component specified by xy select point(s).	
#	Same as +.	
-	Exclude the vertex or component specified by xy select point(s).	
:G	Select an entire collection. :G is used for gathering old collections into a new collection.	
:R	Specifies to include or exclude the vertices of a rectangle, polygon, and line vertices. The vertices are enclosed in a rectangular region defined by two xy points.	
:T	Specifies that an operation will be performed on the gathered collection. :T must be followed by one or more of the following: :A, :R, :X, :MX, :MY, and/or :MXY.	
:A	Specifies to move the collection .	
:R	Specifies to rotate the collection.	
:X	Specifies to scale the collection.	
:MX[1]	Specifies to mirror the collection about the X axis.	
:MY[1]	Specifies to mirror the collection about the Y axis.	
:MXY[1]	Specifies to mirror the collection about both the X and the Y axes.	
:U	Ungather the specified collection .	

(Table Continued)

1 Collections cannot be mirrored about an isometric axis.

Item	Description/Default	Range Restrictions
Parameters		
collect name	Unique name of the collection.	8 characters maximum
xy point	xy locations of the two opposite corners of the gathering rectangle.	
xy gather select point	xy point near the collection to be adjusted.	
collect name	Name of the collection to be adjusted.	8 char maximum
component	One of 12 defined descriptors for limiting the search.	A,C,D,E,H,I,L,M,N, P,R,T (See the *Components* section for descriptor names.)
layer	Layer number for limiting search.	1-255
instance name	Name of an instance for limiting the search.	
xy select point	XY location of the vertex or component you wish to include or exclude. Entering additional xy select points allows additional components or vertices to be included or not included in the collection you are creating.	
xy ref point	Reference point for collection.	
xy place point	New xy location for xy reference.	
rotation angle	Angle, in degrees, you wish to rotate the collection. The angle is measured from an imaginary horizontal line.	
scale factor	Amount by which you wish to change the scale of the existing collection.	

Use of GATHER

You can use GATHER to perform one or more of the following operations:

- Build a collection of components and/or vertices
- Perform an operation on a collection (with :T and a subsequent option).
- Ungather a collection.

You can use GATHER to build a collection in one of two ways:

- Enter `GAT` and enter or select the xy points which define the opposite corners of the collection rectangle. All components included within this rectangle become highlighted until the command is terminated.
- Enter `GAT +` and select individual components. The collection will contain only those components selected.

You cannot edit a complete gathered collection. However, components and/or vertices can be included in and/or excluded from a collection by one of the following four ways:

- Use the + or − options to specify particular components to include or exclude from the collection *before* terminating GATHER. The current snapping mode determines the effect of the + and − options. If vertex snapping is in effect, individual *vertices* are included or excluded. If any other snapping mode is in effect, individual *components* are included or excluded.
- Wrap or delete a component, a portion of a component, or a vertex to remove it from the collection.
- Make a vertex colinear with another vertex of the same component and the vertex is removed from the collection.
- Use GATHER to create a new collection which includes or excludes the components you want in a group.

Graphics Editor commands (such as MODIFY, MOVE, COPY, DELETE, WRAP, and ARCHIVE) cannot be used on an entire gathered collection. However, these commands can be used on individual components of the collection.

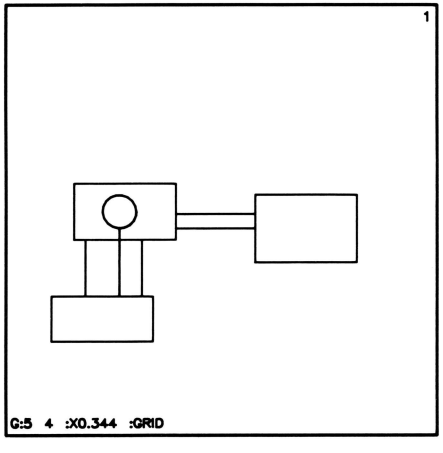
Although you cannot gather individual vertices of an instance, you can gather an entire instance as part of a collection. If an instance within a collection is smashed, each component making up that instance becomes a part of the collection.

If you have not terminated a gathered collection, you can ungather it by entering `:U;`.

Example

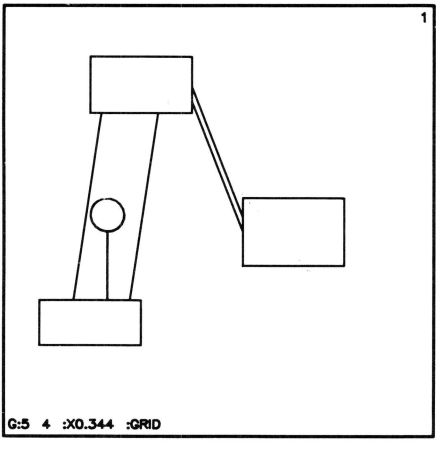
```
ADD R1 :GRI -150,-41 -60,9 -171,-131 -80,-90 9,-61 99,0;
ADD L1 :PRI -90,-40 -90,-91;
ADD L1 -140,-40 -140,-89;
ADD L1 -61,-30  10,-30;
ADD L1 -61,-17 10,-17;
ADD C3 -111,-17 -110,-30;
ADD L3 :PRI -110,-30 -110,-90;
GAT BOX3 :GRI -175,-65 -20,35 :PRI -C3 -110,-30 -L3 -110,-30 :T   -150,-40
:A :GRI -125,75;
```

This example first adds several components with the `ADD` commands. These components appear below:



The GATHER command creates a collection called `BOX3`. This collection is bounded by the rectangle formed by the xy points (-175,-65) and (-20,35). All of the components (including connecting lines) within this rectangle are part of the collection except the circle and line on layer 3. The circle is excluded by specifying `-C3` and a point on its outline (-110,-30) and the line is excluded by specifying `-L3` and the same point.

The collection is moved by specifying :T, an xy reference point on the collection (-150,-40), :A, and the xy place point (-125,75).



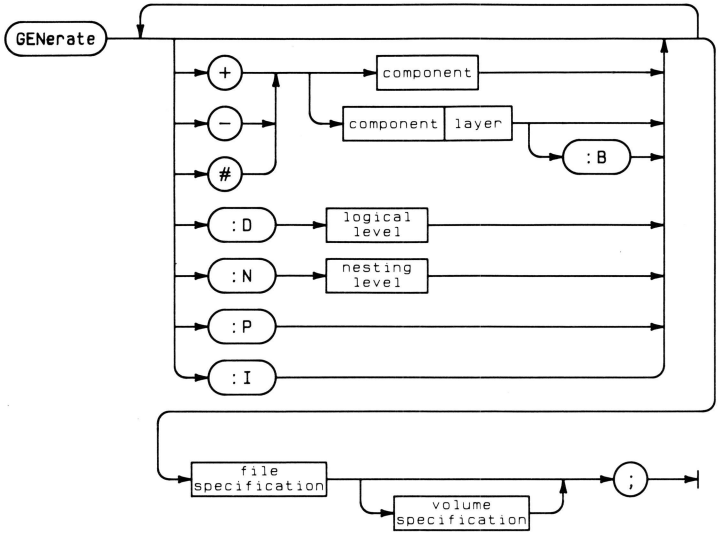
Notice that the circle and line on layer 3 are not moved since they are not part of the collection. Also notice that an xy gather select point did not need to be entered after :T. This is because GATHER remained active after the collection was formed. If GATHER had been terminated by another primary command, the collection could be moved by entering:

```
GAT :T -150,-40 -150,-40 :A -125,75;
```

In this command (-150,-40) served as the xy gather point and the xy reference point.

GENERATE

GENERATE provides a list of the Graphics Editor commands used to create a drawing. Unlike ARCHIVE, GENERATE does not preserve nesting levels and smashes all instances to their original components. GENERATE is a primary command.



Item	Description/Default	Range Restrictions
Options		
+	Specifies to include in the GENERATE file only the components defined by a descriptor. The default is all components on all layers.	
–	Specifies to exclude in the GENERATE file all components except those defined by a component descriptor.	
#	Same as +.	
:D logical level	Specifies to output the GENERATE file so that drawings less than the given logical level are not output. The default is –32768.	–32768 to 32767
:N nesting level	Specifies to include components displayed on nesting levels 1 to the specified depth. The default is all nesting levels. If the drawing contains instances, all instances in nesting levels 1 to the specified nesting level depth are included but smashed into their original component parts. Instances in nesting levels deeper than the specified nesting depth are referenced in the GENERATE file but are no longer contained in the drawing.	
:P	Specifies to place the current process file at the beginning of the GENERATE file. This insures that a called-up GENERATE file will have the correct layer definitions.	
:I	Specifies that implicitly stepped commands are explictly stepped in the GENERATE file.	
:B	Specifies that all layers bound to the affected layer are also generated.	
Parameters		
component	One of the 13 defined descriptors. The default is all components on all defined layers.	A,C,D,E,H,I,L,M,N, O,P,R,T (See the *Components* section for descriptor names.)
layer	Layer containing the component(s) you wish to include in or exclude from the generate file. The default is all components on all defined layers .	0-255
file specification	The name of the GENERATE file.	
volume specification	Volume you wish the generate file stored in.	

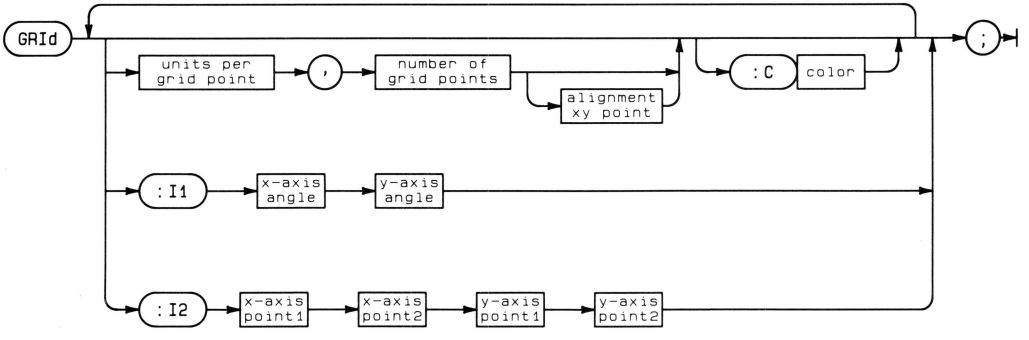
Example

```
GEN -E +L2 +A :N3 EWSYS:part1 EWBEAR:
```

This example takes the drawing file called `part1` from volume EWSYS and creates a GENERATE file with the same name in volume EWBEAR. `-E` specifies to exclude all components. However, `+L2` specifies to include the commands for creating the lines in layer 2. `+A` specifies to include the arcs in all layers. These lines and arcs must reside on nesting levels 1, 2 or 3 .

GRID

GRID determines the coarseness of the user grid, the alignment and orientation of the grid, and the portion of the grid that is displayed (the display grid). GRID is a secondary command.



Item	Description/Default	Range Restrictions
Options		
:I1	Specify an isometric grid by entering the orientation angles of the logical X and Y axes. Enter two angles to define the orientation of the logical axes with respect to the screen x-axis.	
:I2	Specify an isometric grid by entering 4 points that define the logical X and Y axes. :I2 is often used when the angles are not known . Define each axis by selecting two points on the screen.	
:C color	Specifies the grid color. The first seven colors may be specified by either a letter or a number. Colors 8 through 15 are specified by a number. W White 1 R Red 2 Y Yellow 3 G Green 4 C Cyan 5 B Blue 6 M Magenta 7 Black 8 Olive Green 9 Aqua 10 Royal Blue 11 Maroon 12 Brick Red 13 Orange 14 Brown 15 The default color is green on color CRTs and white on monochrome CRTs.	

(Table continued)

Item	Description/Default	Range Restrictions
Parameters		
units per grid point	Number of user units between consecutive user grid points. This is the user grid. Whenever the :GRID mode is in effect, xy locations snap to the user grid. When attempting to align a new grid with a current user grid point, it may be necessary to change the user grid spacing to allow alignment with the xy location that you want. User units are specified in the process file.	
number of grid points	Defines how often the grid points are displayed. This is the display grid.	
alignment point	Shifts the grid such that a grid point is located exactly on the alignment point. The default alignment point is the drawing origin 0,0.	
x-axis and y-axis angle	Specifies the orientation angles of the logical x and y axes.	
x-axis point1, x-axis point2, y-axis point1, y-axis point2	Specifies the 4 points that define the logical X and Y axes.	

Hints

When an isometric grid is used, the cursor is changed to show the orientation of the axes. It is often convenient to use an oversize cursor (or full screen cursor) to help visualize the orientation of the axes.

NOTE

After the isometric grid is turned on, all subsequently added circles are converted to ellipses (ovals). Arcs are converted to sections of ovals and rectangles are converted to regular polygons. Ovals remain as ovals.

The isometric grid may be converted back to the normal grid by entering:

```
GRI   :I1 0,90 ; or EDI ;
```

Example

```
GRI  10,2 ;
```

This example sets a new user grid which has 10 user units between each user grid point and displays every 2nd user grid point. If the current user units are mm, then distance between two user grid points is 10 mm. A grid point would be displayed every 20 mm.

Example

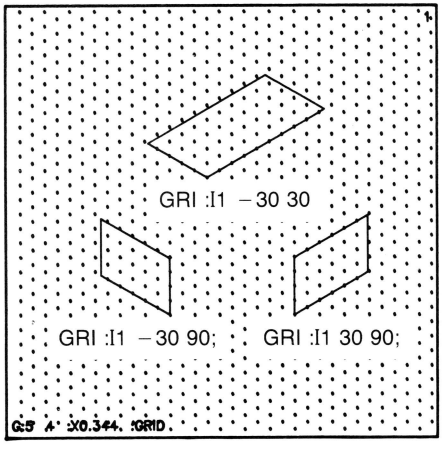
```
GRI :I1 -30 90  ;
```

This example defines a normal isometric grid so that you could draw points on the front face of an object. To draw points on the side face of such an object, enter

```
GRI :I1 30 90;.
```

The top face could be accessed with

```
GRI :I1 -30 30 ; .
```



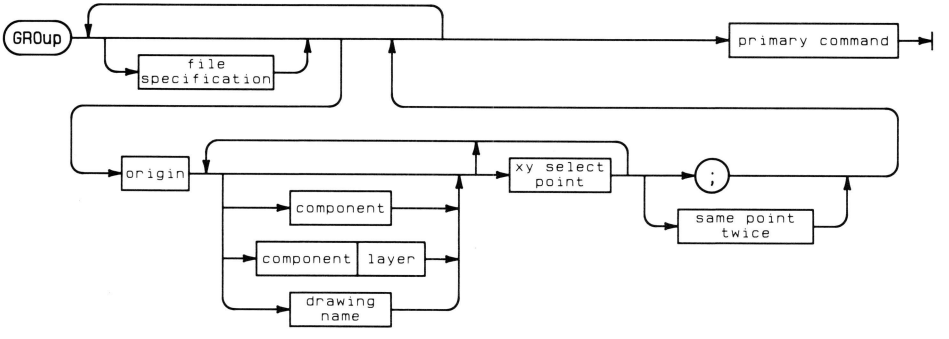
Example

```
GRI ;
```

This example displays the current grid default.

GROUP

GROUP creates a new drawing from components on the screen. A copy of the new drawing is placed in computer memory. The components forming the new drawing are displayed in nesting level 2 and are replaced by an instance on nesting level 1. This instance is linked to the existing drawing. WRAP performs the same functions as GROUP. GROUP is only included to maintain compatability with previous HP EGS versions. GROUP is a primary command.



Item	Description/Default	Range Restrictions
Parameters		
file specification	Name assigned to the GROUP instance. If named, the GROUP instance can be saved as a library part. The name cannot resemble a command or macro name.	8 characters maximum
origin	XY location specifying the origin of the GROUP instance.	
component	One of 13 defined descriptors. This allows you to limit component selection. If the component lies within an instance, you can add the instance name to further limit selection. The entered descriptor is only valid for one selection.	A,C,D,E,H,I,L, M,N,O,P,R,T (See the *Components* section for the descriptor names.)
layer	Layer containing the selected components.	1-255
drawing name	Name of the instance component that you wish to select. If used with a component descriptor, component selection will be limited to the instance. Additionally, assigning a name to the drawing allows an instance of the drawing to be added to other sections of the drawing with the ADD command.	8 characters maximum
xy select point	XY location of the component you wish to select. Entering additional xy select points allows additional components to be included in the drawing you are creating.	

Hints

It may be advantageous to name a GROUP instance because it can be later referenced and placed anywhere on your drawing with `ADD instance`. In addition, a named instance can be saved and referenced to any other drawing.

The following rules apply when using the GROUP command in conjunction with associated text:

- If you pick a component with associated text, you will get the component and all associated text.
- If you pick a Text/Note component which is associated with some other component (:AC), you will get the other component and all of its associated text.
- If you pick a Text/Note component which is associated with a drawing (:AD), you will not be allowed to pick it.

Example

```
GRO group1  50,50  L 60,75  C2  90,80  100,120  ;
```

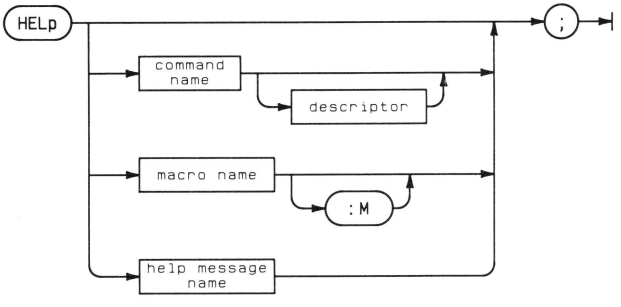
This example creates a GROUP instance called `group1` which has its origin at (50,50). The instance replaces the following components:

- A line which passes through the location (60,75) (only a line component could be selected because the descriptor `L` is specified before the xy point is entered).
- A circle on layer 2 which contains the point (90,80) (only a circle component on layer 2 could be selected at this location because of the descriptor designation and layer).
- The drawing located at the point (100,120).

The instance group1 appears on nesting level 1. The components forming `group1` are displayed in nesting level 2.

HELP

HELP provides detailed information about a command or macro . HELP is a secondary command.



Item	Description/Default	Range Restrictions
Option		
:M	Turns off macro definition.	
Parameters		
command name	Name of the command you want information on.	
descriptor	One of the 11 defined descriptors	A,C,D,H,I,L, M,N,P,R,T (See the *Components* section for descriptor names.)
macro name	Name of a currently defined macro. The program will first briefly explain the operation of the macro and then display the definition of the macro.	20 chars max
help message name.	A string of characters which defines a help message name. This might be a help message you have defined and placed in the current help file. The HELP file is loaded with `LOA HELP filename ;`	20 chars max

Hints

If you enter `HEL ADD ;`, the system presents a general discussion of ADD. If you enter `HEL ADD descriptor ;`, where `descriptor` is one of the 11 defined descriptors, the system presents a detailed discussion of the methods used to add the selected component.

If you enter `HEL ;`, a general description of how to use the help command itself is presented.

The syntax appearing in the help messages uses the following notation.

- Items enclosed in square brackets (`[ ]`) are short notes describing optional parameters or data.
- Items enclosed in braces (`{ }`) represent a choice of items, where at least one of the items must be entered.
- Items separated by a bar (`|`) are choices of parameters, data, or items to enter with a command.

If you select `HELP*` and a command from the screen menu of one of the HP EGS personalities, the help message will explain the personality command. In some cases this command will be different than similar looking system-level commands.

Example

```
HEL ADD A   ;
```

This example displays the following message on the CRT:

```
ADD Arc adds an arc to the existing drawing.

   ADD A[layer] [:R res] [:W width] [:FILL] {:P3|:A|:C|:T2} xy1 [xy2,,,] ;

EXAMPLE:   ADD A4 :R 20 :W 20 :FILL :A 50 120 240 100,200;

This example adds an arc on layer four.  The arc has a resolution of 20
and has a filled width of 20.  The  radius is 50, its starting angle 120,
its finish angle 240, and its center point is at 100,200.

COMMAND
   A       The descriptor specifying an arc component.
   layer   Layer you are adding the arc to.  Range is 1-255 layers.
OPTIONS
   :R res  The resolution angle  (in degrees) between endpoints of the line
           segments which form the arc.  Default is 0. Range is 0-120.
   :W      Specifies the width of the arc outline in user units.
   :FILL   Specifies to fill the outline of the arc.
   step spec.    Adds an implicit component step.  See ADD for explanation.
```

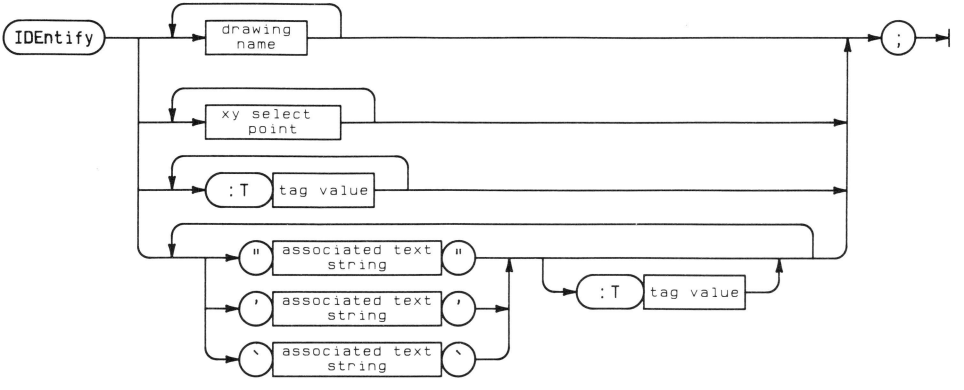
(Example Continued)

```
METHODS
   :P3   Enter the  two endpoints of the arc and a point on its circumference.
         This is the default method.
   :A    Enter the  radius, the angle to its starting point, and the
         angle to its end.  The arc extends between these two angles.
   :C    To draw a fillet with an arc.  Enter the  radius and the vertex
         formed by two intersecting segments.
   :P2   Enter the  arc's two endpoints. A semi-circle will connect the points.
   :T2   Enter the arc's radius value, two tangent points to a component and the
         arc's approximate center point.  The arc connects the tangent points.
PARAMETERS
   xy endpt       Endpoint of the arc.
   xy circum. pt  A point on the circumference of the arc.
   radius value   Radius of arc.
   starting ang.  Angle (in degrees) measured counterclockwise from a horizontal
                  to the first arc endpoint.
   fin angle      Angle (in degrees) measured counterclockwise from a horizontal
                  to the second arc endpoint.
   center point   Center of arc.
   xy corner pt.  Defines a vertex formed by two intersecting line segments.
   xy tang. pt.   The point where the arc is tangent to a component.
   approx.cnt pt  The estimated center point of an arc tangent to components.
```

The help message syntax shows that everything is optional except the arc descriptor and xy1 and xy2.

IDENTIFY

IDENTIFY identifies any geometry, instances or associated text/notes which you select on the CRT display. IDENTIFY is a secondary command.



Item	Description/Default	Range Restrictions
Option		
:T tag value	Tag value of text, notes, and/or components to be identified. Once selected, all text, notes, or components of the given tag value are highlighted.	-32768 to +32767
Parameters		
drawing name	Instance to be identified. This highlights all instances with the specified name.	
xy select point	XY point near component to be identified. Once selected, the component is highlighted and its attributes are listed at the top of the screen.	
associated text string	String value of associated text to be identified. Once selected, the string is highlighted along with associated components.	

Example

```
ADD T "TEXT" :F 30 :S 30 30,0;
IDE 30,0;
```

This example first adds text to the screen. The IDENTIFY command identifies the text by printing at the top of the screen:

```
Text 1 :F30 :R0.0000 :S30.0000
```

This identifying line states that the component is text on layer 1. The text has a font of 30 and slant of 30°.

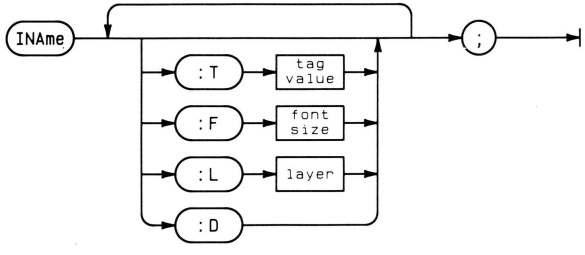
Example

```
IDE NAND ;
```

This example identifies all instances with the name `NAND`.

INAME

INAME transfers INAME associated text created in EGS-45 to the HP EGS system. INAME is a secondary command.



Item	Description/Default	Range Restrictions
Options		
:T tag value	The default tag value for INAMES entered from EGS-45 archive files.	
:F font size	The default font size for INAMES entered from EGS-45 archive files.	
:L layer	Sets the default layer for INAMES entered from EGS-45.	1-255
:D	Specifies that INAMES entered from EGS-45 archive files will be displayed.	

Hints

When an archive file created by EGS-45 is transferred to HP EGS, the INAME associated text is converted to HP EGS associated text. Because HP EGS associated text has some parameters which are not part of the archive file for INAME (tag value, font size, layer and display option), default values for these parameters must be defined. These default values are defined with the INAME command.

The default values are stored in the process file and may be changed at any time.

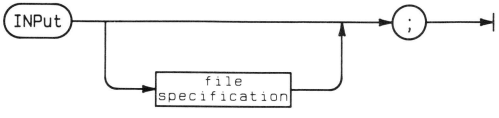
Example

```
INA :T 1006 :F 3 :D :L 7 ;
```

This example transfers text from EGS-45 . The text is given a tag value of 1006, a font size of 3 and is displayed on layer 7.

INPUT

INPUT opens a file containing a sequence of commands. You use the Pascal Editor to create this sequence of commands or use GENERATE to list the commands for creating a drawing. RETRIEVE is usually used to reinput ARCHIVE files. INPUT is a secondary command.



Item	Description/Default	Range Restrictions
Parameter		
file specification	File to be input.	

Hints

The input file cannot contain *nested* INPUT or RETRIEVE commands (such as commands which change the current file).

Press STOP to stop the reconstruction of a GENERATE drawing file. When you press STOP, the system asks if you really want to stop the input process. If you answer Y for yes, the system returns control to the keyboard or graphic input device. If you answer N for no, the input process continues.

To re-begin the input process, enter: `INPUT ;` and the input process continues from the point where you pressed STOP.

If an error occurs during the input process (such as an improper command in the generate file), one of two things will happen:

- If ECHO is *disabled*, the system transfers control to the keyboard or graphic input device. You may remedy the error by correctly entering the erroneous command on either the keyboard or the graphic input device. Then enter: `INPUT ;` to resume the input process.

  If ECHO is *enabled*, the error is printed on the printer and the input process continues. You may then use the commands of the Graphics Editor to correct the error when the input process is finished.

For additional information on stream files and how to input them see the *Customizing Your System* chapter of *Understanding HP EGS*.

Example

```
INP EWSYS:gdstrt.TEXT ;
```

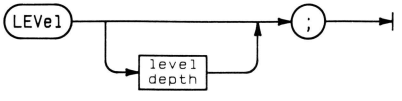
This example inputs the file gdstrt.TEXT. This file contains the following sequence of commands:

```
loa com load;
loa com define;
loa com menu;
loa com window;
loa com add;
loa com delete;
loa com modify;
loa com stretch;
loa com move;
loa com gather;
loa com copy;
loa com step;
loa com wrap;
loa com smash;
loa mac gd;
loa hel gd;
loa pro gd;
loa sea hpegs;
loa tmenu gd;
loa dra l_arrow;
loa dra d_arrow;
loa men gd;
initialize_gd
```

LEVEL

LEVEL displays or changes the level number of the drawing currently being edited. It is primarily used with LIST MATERIAL, ARCHIVE, and GENERATE. ARCHIVE and GENERATE have a :D option which allows you to specify the level depth to which a drawing is to be processed. If :D is not specified, then logical levels are ignored during the processing.

The default logical level is 1 for drawings. Instance levels must be less than or equal to the drawing level number. For example, HP EGS library parts are at logical level 0. LEVEL is a secondary command.



Item	Description/Default	Range Restrictions
Parameter		
level depth	The level depth to set the current drawing to.	-32768 to 32767

Example

```
LEV 5 ;
```

This example sets the current level depth to five.

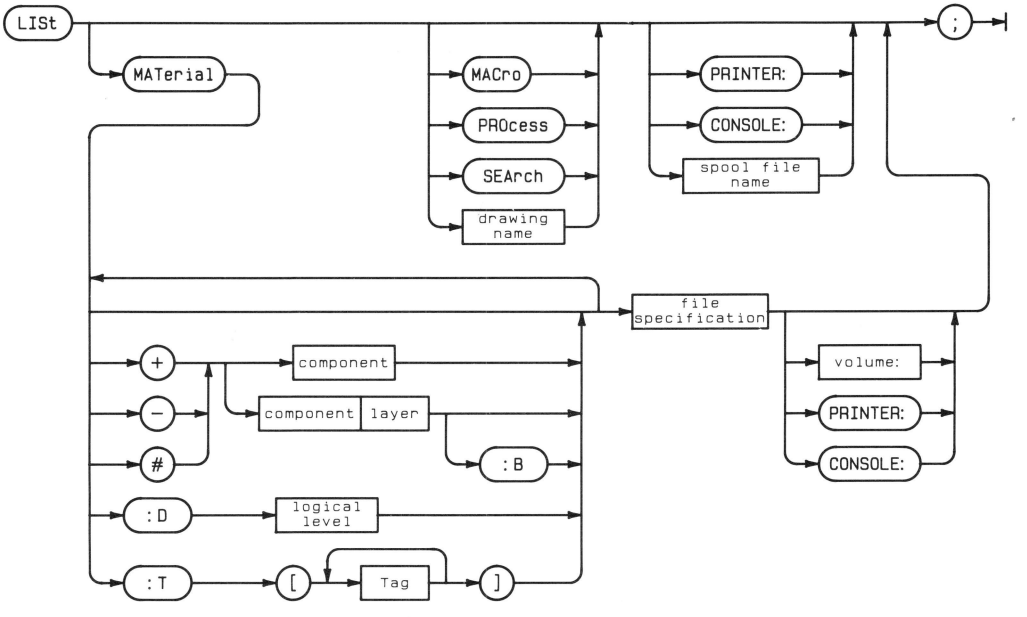
Example

```
LEV ;
```

This example displays the current level depth.

LIST

LIST lists the macro, process and drawing files currently loaded in the system, describes any drawings currently in computer memory, and makes material lists and connection lists directly from the Graphics Editor. In addition, LIST information can be routed to a printer, SRM spool file, or the alpha screen. LIST is a primary command.



Item	Description/Default	Range Restrictions
Commands		
MACro	Specifies to list the contents of the current macro file. A macro file has the suffix `_mc,TEXT`.	
PROcess	Specifies to list the contents of the current process file. A process file has the suffix `_pr,TEXT`.	
SEArch	Specifies to list the contents of the current volume table used for searching. A volume search table has the suffix `_vs,TEXT`.	
MATerial	Specifies to prepare a file containing a material list of the drawing. A material list output has the suffix `_m`.	
Options		
+	Specifies to include a particular component in the list.	
−	Specifies to exclude a particular component from a list.	
#	Specifies to include all components in the list.	
:D	Smashes instances greater than the level number specified.	
:T	Specifies to list the specified text/note components associated to part instances. Any tag fields chosen in this manner appear in additional columns on the right-hand side of the listing.	
:B	Specifies that any component on a layer bound to a selected layer is to be considered physically and logically connected to the components on the bound layer.	
PRINTER:	Specifies to output the list to the printer rather than the graphics screen.	
CONSOLE:	Specifies to output the list to the alpha screen rather than the graphics screen.	

Item	Description/Default	Range Restrictions
Parameters		
drawing name	Name of the drawing you want to list. The list specifies how many of each type of component is used in the drawing, as well as all drawings linked to that drawing. CAUTION: Do not use `_d` in the name as this overwrites the drawing file (the system places the list of drawing components in that file).	
spool file name	Name of file list is to be sent to.	SRM spool files must end in .ASC
logical level	Level to which you want parts to be hierarchically netted.	−32768 to +32767
tag value	Tag value of the associated text components you wish to list.	−32768 to +32767
component	One of 13 defined descriptors.	A,C,D,E,H,I,L,M, N,O,P,R,T (See the *Components* section for descriptor names.)
layer	Layer containing components you wish to list.	
logical level	Level to which you wish to SMASH instances.	
file specification	Specifies the drawing file you wish to process.	
volume specification	Specifies volume where the output list will be sent.	

Additional information about material and connection lists appears in *Understanding HP EGS*.

Example

```
LIS;
```

This example causes the system to print a list of all macro, process and drawing files currently loaded in the system.

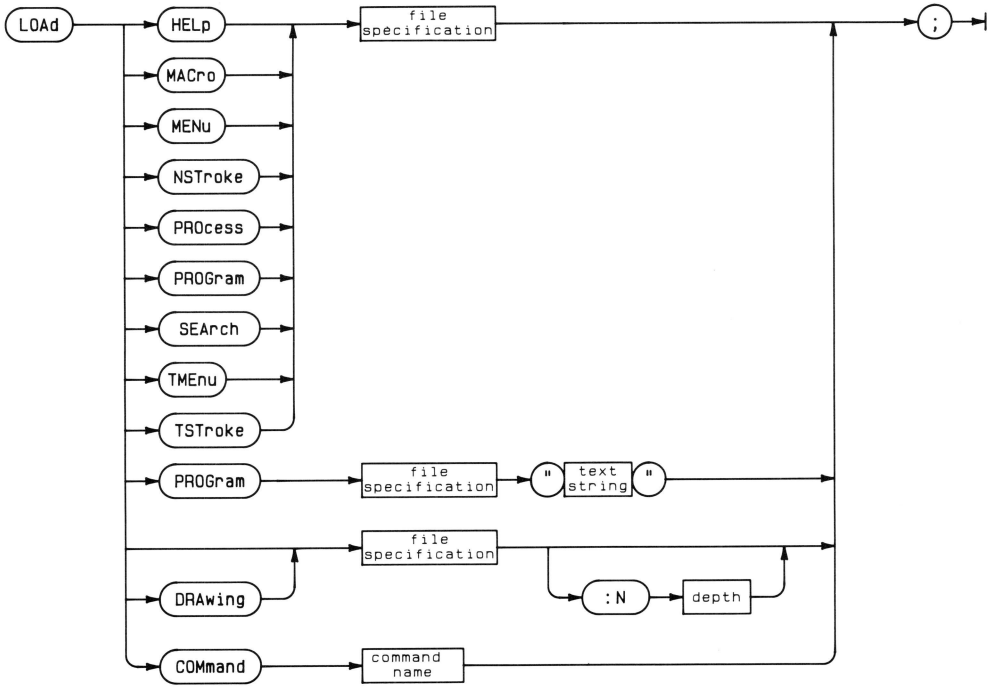
Example

```
LIS MAC ;
```

This example causes the system to print a listing of the current macro file.

LOAD

LOAD loads files of any type into the system. LOAD is a primary command.



Item	Description/Default	Range Restrictions
Commands		
HELp	Specifies to load a second message file (in addition to EWCODE:gedit_ms.ASC) into the system. The second message file might contain help information not defined by EWCODE:gedit_ms.ASC or current HP EGS module help files. NOTE: You can only load one help file at a time; therefore if you load a new help file you will overwrite the current one.	
MACro	Specifies to load a macro file. A macro file has the suffix _mc.TEXT.	
MENu	Specifies to load a menu file. A menu file has the suffix _mn.TEXT.	

Item	Description/Default	Range Restrictions
Commands (cont.)		
NSTroke	Specifies to load a new stroke table for notes. A stroke file has the suffix `_st,TEXT`.	
PROcess	Specifies to load a process file. A process file has the suffix `_pr,TEXT`.	
PROGram	Specifies to load another PASCAL program from the Graphics Editor. While loading, the recall buffer is erased and the Graphics Editor is temporarily suspended. When the system finishes loading, it returns to the Graphics Editor and turns on the graphics. The type ahead buffer can be used by entering a string expression after the file specification. NOTE: The system has about 13 Kbytes of available global space for loading programs. There is not enough global space to run HP TechWriter from the Graphics Editor.	
SEArch	Specifies to load a volume search file. A volume search table file has the suffix `_vs,TEXT`.	
TMEnu	Specifies to load a tablet menu file. When loading the tablet menu, the system checks to see if the four corner coordinates are loaded. If so, the system asks if you wish to use them. If you do, the tablet menu is ready for use. If the four corner coordinates are not loaded, the system prompts you to place the tablet menu on the tablet and digitize the corners. The tablet menu is then ready for use and will remain loaded until you leave the Graphics Editor or load a different tablet menu. A tablet menu file has the suffix `_tm,TEXT`.	
TSTroke	Specifies to load a new stroke table for text. A stroke file has the suffix `_st,TEXT`.	
COMmand	Specifies to load a segmented command into memory. The command will stay in memory until you quit the current drawing session.	
DRAwing	Specifies to load a drawing file. A drawing file has the suffix `_d`.	

Item	Description/Default	Range Restrictions
Option		
:N	Specifies to load a drawing to the depth specified.	Valid only with LOAD DRAWING.
Parameters		
command name	The command name for any HP EGS command.	
file specification	Name of file to be loaded.	8 characters maximum for a drawing, 6 characters maximum for all other EGS files.
text string	Text put in the type-ahead buffer and passed on to the next program.	8 characters less than screen width.
depth	Load the drawing such that components and instances at this value and higher are brought into memory.	

Loading Commands

Each HP EGS command is segmented; that is, each command is stored as an individual file in the volume EWCODE. Because of this, each time you wish to use a command it must be available in current memory or be temporarily loaded from the disc. Many of the common commands are automatically loaded into memory by the start files of the HP EGS modules. The LOAD command also permanently loads commands into memory. Once loaded, a command remains in memory until you quit the current drawing session.

Each command requires a certain amount of memory space. If you working on a drawing or performing another task that requires a large amount of memory, you may wish to preserve the memory by limiting the number of loaded commands. However, in contrast to instantaneous access of commands stored in memory, the system takes a certain amount of time to temporarily load a command from the disc. The amount of time depends on the size of the command; for small commands, the time delay is insignificant. For larger commands (such as ADD) the time delay is much longer. Therefore, you must decide which is more important: the time delay during temporary loading of a command from the disc or or the memory loss caused by a command already loaded into memory. You can estimate the amount of memory a command would use by looking at the size of the command code file in the volume EWCODE.

If you are using an HP EGS module, you can alter the module's start file to specify the commands you want loaded into memory.

Example

```
LOA PRO genrl   ;
```

This example loads the process file genrl_pr.TEXT .

Example

```
LOA TME  specl ;
```

This example loads the tablet menu specl_tm.TEXT

Example

```
LOA part1       ;
```

This example loads the `part1`. If the file type is not specified by a suffix, the system assumes that the desired file is a drawing file.

Example

```
LOA PROG *FILEREGS ;
```

This example loads the Pascal Filer. As the Filer is loaded, the recall buffer is erased. (The Filer specification depends on the way the SRM is configured.) To leave the Filer and return to the Graphics Editor, enter `Q` for Quit.

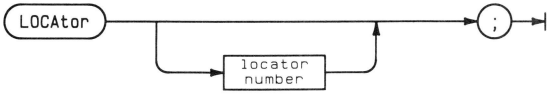
Example

```
LOA COM add;
```

This example loads the ADD command. After a segmented command has been loaded, the system does not need to re-load it each time you use it. The command is removed when you quit the current drawing session.

LOCATOR

LOCATOR selects the graphic input device. LOCATOR is a secondary command.



Item	Description/Default	Range Restrictions
Parameter		
locator number	Line number of a line in the EWCODE: gedit_ms.ASC message file associated with the LOCATOR tag. The LOCATOR tag is several lines listing the possible graphic input devices. If you specify a locator number greater than the number of devices defined in the tag, an error message is displayed and the command is ignored.	1 through the number of lines in the locator tag.

Describing Input Devices

The LOCATOR tag in the EWCODE:gedit_ms.ASC message file appears as:

```
LOCATOR      "D"706 HPGL
             "D"0   HIL_MOUSE
             "D"0   HIL_TABLET
             "D"9   RS232_TABLET
```

The HIL_MOUSE is the HP 46060A HP-HIL Mouse. The HIL_TABLET is the HP 46087A HP-HIL A-size digitizer or the HP 46088A HP-HIL B-size digitizer. The RS232_TABLET is connected to the computer by either the HP 98626A RS-232C Serial Interface Card or the HP 98644A RS-232C Serial Interface Card.

Example

```
LOCA ;
```

This example displays the current input device and its address.

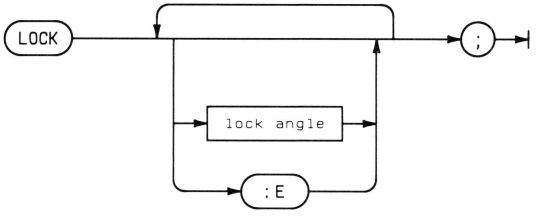
Example

```
LOCA 2 ;
```

This example selects the HP 46060A HP-HIL Mouse from the list above. This example uses number 2 since the HIL_MOUSE is on the second line of the locator tag.

LOCK

LOCK determines the value for lock angle warning. The system issues the warning when the angle of a component (such as the angle of a line) is not an integer multiple of the lock angle. This warning is not a critical one but rather it is a message to aid you when entering components. The lock angle is always measured from the horizontal x-axis regardless of the isometric grid setting. LOCK is a secondary command.



Item	Description/Default	Range Restrictions
Option		
:E	A toggle which enforces lock angle snapping. When lock angle is enforced, the system will try to snap according to the current snapping mode. If such snapping does not produce the given lock angle value or a multiple of it, the system will display the lock angle warning and the component will remain unchanged. The lock angle is always enforced from the horizontal x-axis regardless of the isometric grid setting. Setting the lock angle to zero turns the warning message off. The default is 0° with enforcement turned off.	
Parameter		
lock angle	The lock angle, in degrees, that you wish to set. Setting the lock angle to zero turns the warning message off.	0-180° in .0001° increments

Use of LOCK angle

Lock angles are very helpful for certain applications. For example, when laying out traces for a printed circuit board, the angle formed by the intersection of two traces (line segments) is usually 45°. The lock angle warning would notify you whenever the angle is not exactly 45° (or an integer multiple of 45°).

The lock angle is only checked and enforced when a component is being added, stretched, or rotated. If an instance is added, the instance is not checked to insure that each of its components correspond to the lock angle. Likewise, if something is added when lock angle checking is disabled, those components will not be re-checked should lock angle checking be activated. Enforcement does not apply to mirroring operations.

Example

```
LOC 30;
```

This example sets the lock angle to 30°.

Example

```
LOC 90 :E ;
```

This example sets and enforces the lock angle to 90°. Lock angle enforcement would be toggled off if you entered `LOC :E;` after the previous command.

Example

```
LOC ;
```

This example displays the current lock angle and the status of lock angle enforcement.

MEMORY

MEMORY displays the amount of memory in the "never used" block. The "never used" block is the memory remaining after the Graphics Editor program has been loaded. As components are added, this "never used" block is gradually used for storage. If a component is deleted, the memory it occupied is recycled and reused before any new memory is taken from the "never used" block. Thus deleting a component will not increase the memory displayed by the MEMORY command. MEMORY is a secondary command.

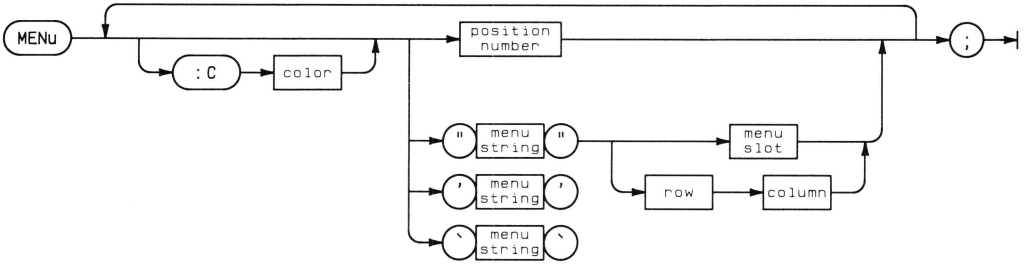
Example

```
MEM  ;
```

This example displays the amount of remaining memory.

MENU

MENU adds or deletes entries to the existing menu. MENU is a secondary command.



Item	Description/Default	Range Restrictions
Command position number	Locates the position of the menu on the screen area.	
Option :C color	Specifies the color to display the string being added to the current menu. The first seven colors may be specified by either a letter or a number. Colors 8 through 15 are specified by a number. An equal sign specifies the inverse color (for example, `:C=R` will display inverse red). W White 1 R Red 2 Y Yellow 3 G Green 4 C Cyan 5 B Blue 6 M Magenta 7 Black 8 Olive Green 9 Aqua 10 Royal Blue 11 Maroon 12 Brick Red 13 Orange 14 Brown 15 The default color is green on color CRTs and white on monochrome CRTs.	

(Table continued)

Item	Description/Default	Range Restrictions
Parameters		
menu string	The set of characters you wish to add to the menu. By entering an empty string (`MENU " "`) in a menu box, you erase an existing entry.	
menu slot	The menu slot you wish to place the menu string in. It must be picked with the tablet stylus. If that menu space is already occupied, the new menu item replaces the old.	
row	The row you wish to place the menu string in. Rows are counted from the top, starting with one.	
column	The column you wish to place the menu string in. Columns are counted from the left, starting with one.	

Example:

```
MEN :CR "AREA"
```

This example adds the characters `AREA` to the menu in the menu box you select with the stylus. The characters are displayed in red.

Example

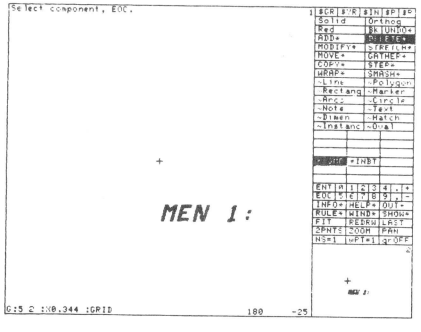
```
MEN :C=G "BOX"  17 2;
```

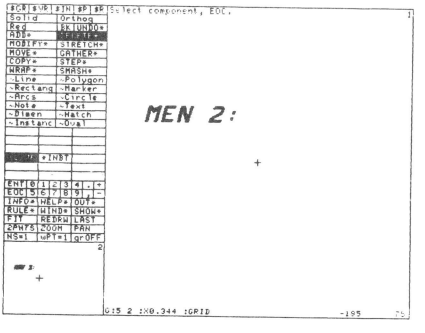
This example adds the characters BOX to the menu. BOX is placed in the seventeenth row and second column and is displayed as inverse green.

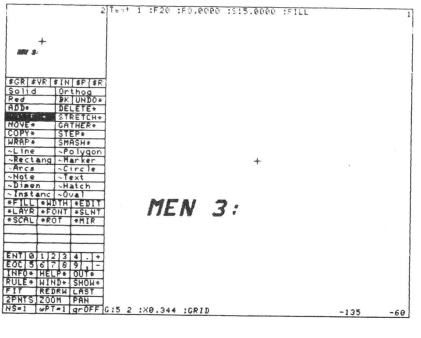
Example

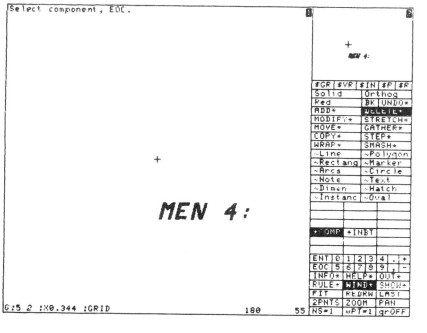
```
MEN 1;
```

This example locates the menu in the upper right corner of the drawing area.



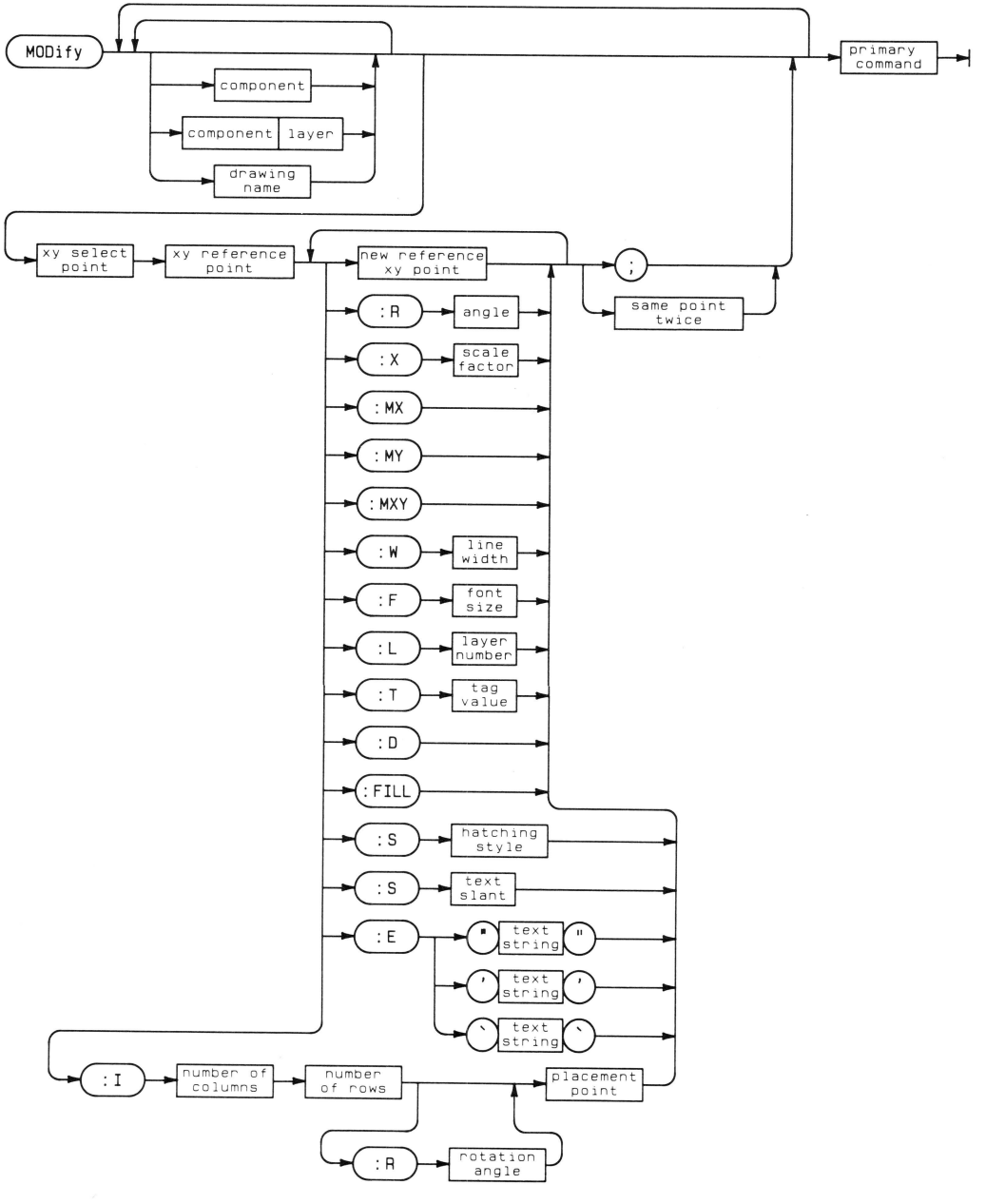






MODIFY

MODIFY changes existing components on the screen. These changes include rotating, mirroring, scaling, font size, line width, changing layers, moving components, changing the step specification, and editing text. When modifying a component, its parameters are displayed in the upper left corner of the screen. MODIFY is a primary command.



Item	Description/Default	Range Restrictions
Options		
:R angle	Specifies to rotate the component. The angle is measured conuterclockwise from an imaginary horizontal line. When both a rotation and a mirror are specified, the mirror is always done before the rotation.	−360° to +360°
:X scale factor	Specifies to change the scale of the existing component when displayed in your drawing. The current units of the component are multiplied by the scale factor to obtain the new scale.	
:MX	Specifies that the component to be modified is to be mirrored about the X axis.	
:MY	Specifies that the component to be modified is to be mirrored about the Y axis.	
:MXY	Specifies that the component to be modified is to be mirrored about both the X and Y axes.	
:W line width	Specifies the width of a line in current user units. MODIFY :W changes hatch width as well as component width.	0 through the number of user units assigned to the 10^9 system grid points.
:F font size	Specifies the height of characters in current units.	1 through the number of user units assigned to the 10^9 system grid points.
:L layer number	The new layer number for the component selected. Only instance components cannot be moved to a new layer.	1-255
:T tag value	Specifies that a tag value is to be assigned to associated text.	
:D	Specifies to toggle the display associated text.	
:FILL	Specifies to fill arcs, circles, lines, polygons, rectangles and/or text.	
:S hatching style	Specifies hatching style. 1 Solid fill 2 Single hatching 3 Cross hatching	1-3
:S text slant	Specifies the text slant.	−75° to +75°

(Table continued)

Item	Description/Default	Range Restrictions
Options (continued)		
:E text string	Specifies that the selected text or note is to be replaced by a new text string..	
:I	Specifies to change the step specification of a component.	
Parameters		
component	One of 13 defined descriptors.	A,C,D,E,H,I,L,M,N, O,P,R,T (See the *Components* section for descriptor names.)
layer	Layer containing the component to be modified.	1-255
drawing name	Name of the instance to be modified.	
xy select point	Point lying in or on the component to be modified.	
xy reference point	XY point identifying the part of the component to be modified.	
new reference xy point	XY point identifying the location to which you want to move the component. Entering additional xy points replaces the location point with each new point entered.	
number of columns	The number of columns in the step array (The step can be removed by entering 0.)	
number of rows	The number of rows in the step array	
placement point	Specifies the distance between reference points.	

Note: When changing the layer number with MODIFY, the color and/or line type will not change until the command is finished.

Hint

MODIFY can be used to find out what layer a component is on or to list the font size, rotation, and slant of text and notes. To do this, enter:

```
MOD
```

and select the component or characters with the stylus. The system will display the information in the upper left-hand corner of the screen area.

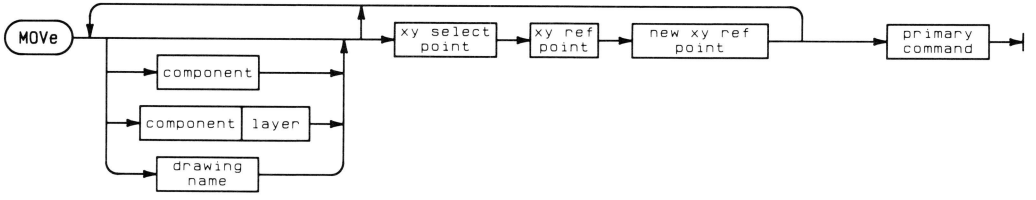
Example:

```
ADD L4 0,0 100,100 100,200;
MOD 0,0 100,100 :X -2 :R 45 :W10 :L3 -100,100;
```

This example first adds a line on layer 4 with vertices at (0,0), (100,100), and (100,120). The second command modifies the line by using (0,0) as the xy select point and (100,100) as the xy reference point. The line is then reduced to half of it original size, rotated 45°, given a width of 10, changed to layer 3 and moved to (-100,100).

MOVE

MOVE moves any component from its current location to a new location on the drawing. MOVE provides a specific subset of the capabilities of MODIFY. MOVE is a primary command.



Item	Description/Default	Range Restrictions
Parameters		
component	One of the 13 defined descriptors.	A,C,D,E,H,I,L,M,N, O,P,R,T (See the *Components* section for descriptor names.)
layer	Layer containing the component to be moved.	1-255
drawing name	Name of the instance that you wish to move.	8 characters maximum
xy select point	Point lying near the component to be moved.	
xy reference point	XY point reference point on the component.	
new xy reference point	XY point of the new location of the component. The xy reference point on the component is placed at this new point.	

Hint

If the component being moved has any text associated with it, the text is also moved.

Example

```
ADD T2 :F15 'TEXT STRING' 10,20 ;
MOV T2 10,20 0,0 -50,-50 ;
```

This example first adds `TEXT STRING` to layer 2 at (10,20). The MOVE command then moves it to (-50,-50).

NOP

NOP returns the system to its original state. It is often used in macros to exit a primary command without using another primary command. NOP is a primary command.

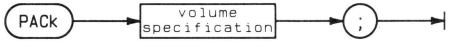
Example

```
ADD NAND2 0,0; NOP
```

This example will turn off dynamic tracking of the instance `NAND2` and terminate the ADD command.

PACK

PACK moves all files on a volume so that all the unused storage space is at the end of the volume. This process becomes necessary when, after repetitive reading and writing to the disc, the available storage space becomes highly fragmented. For example, the disc may have 100 blocks available on the disc, but because they are all in 10 or 15 block chunks, there is not enough contiguous storage space for the system to write a 20-block file to the disc. PACK does nothing on SRM volumes. PACK is a secondary command.



Item	Description/Default	Range Restrictions
Parameter		
volume specification	Volume you wish to pack.	

Note

Before beginning a pack you should backup your disc to insure against data loss from a power failure or other unforseen occurence. UNDER NO CIRCUMSTANCES SHOULD YOU ATTEMPT TO DISTURB THE PACK OPERATION ONCE IT HAS BEGUN. By disturbing the pack process you are risking the loss of the information in the volume. Do not touch the power switch or the door on the disc drive during a PACK.

Example

```
PAC #3: ;
```

This example begins a PACK of volume #3. After entering the above command the system will prompt:

```
Pack directory "V3:" ? (Y/N)
```

If you enter Y for yes, the system will display the message:

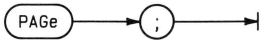
```
Pack of directory "V3:" in progress.  DO NOT DISTURB.
```

Once the PACK is complete the system will display:

```
Pack of directory "V3:" is complete.
```

PAGE

PAGE outputs a top-of-form command to the current system printer. If you are routing your commands to an echo file you might enter `PAG ;` to insure that a screen dump begins on the top of a page form. PAGE is a secondary command.



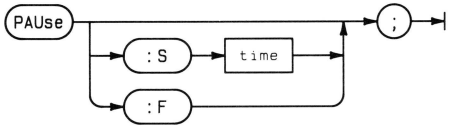
Example

```
PAG  ;
```

This example advances the system printer to the top of the next page.

PAUSE

PAUSE causes the system to pause and request that a key be pressed before continuing. This command is often used to stop or postpone an input file at a particular point. A message is displayed while the system is paused. PAUSE is a secondary command.



Item	Description/Default	Range Restrictions
Options		
:S time	Specifies to pause for the given time (in seconds)	
:F	Starts the pause time.	

Example

```
PAU   ;
```

This example causes the system to pause until a key is pressed.

Example

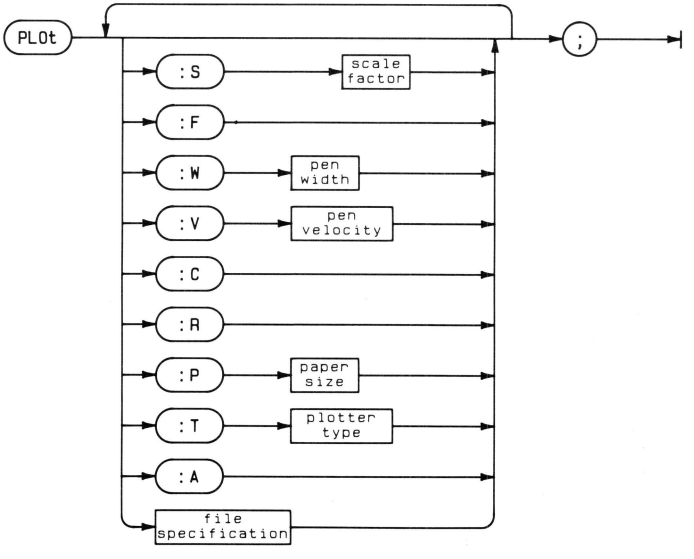
```
PAU :S3600 ;
```

This example might be used for a self-running demonstration. The demonstration input file begins with `PAU :S3600;`. This command would be entered at the top of the hour. Once the input file finished, (in less than 3600 seconds) `PAU :F;` causes the system to wait until the top of the next hour to re-input the demonstration input file.

Entering (STOP) during this waiting period will return control to the user.

PLOT

PLOT sends the current drawing to a plotter or file; the current SHOW* option and nesting level apply when the drawing is sent. PLOT is a secondary command.



Item	Description/Default	Range Restrictions
Options		
:S scale factor	Specifies the drawing scale.	
:F	Specifies to fill in solid all lines with width, rectangles, and circles when the drawing is plotted.	
:W pen width	Width of the plotter pen, in mils. :W causes the plotter to shrink the size of a component slightly to compensate for the pen width. In this way the dimensions of a component are not changed. :W only affects filled lines, filled circles, circles with filled width, arcs with filled width, filled polygons, polygons with filled width and hatching. :W does not affect the plotting of text. The following table converts pen sizes in mm to mils. (HP EGS requires pen sizes to be specified in mils.) Conversion factor: 39.37 mils/mm.	

Pen Type	Size (mm)	Size (mils)
Fiber	.3	11.81
Fiber	.7	27.56
Transparency	.3	11.81
Transparency	.6	23.62
Roller Ball	.3	11.81
Drafting	.18	7.09
Drafting	.25	9.84
Drafting	.35	13.78
Drafting	.50	19.69
Drafting	.70	27.56
Drafting	1.00	39.37

Item	Description/Default	Range Restrictions
Options (continued)		
:V pen velocity	Velocity of the plotter pen in cm/sec.	Must be an integer
:C	Specifies to fit and center a drawing between P1 and P2 on the plotter. If a scaling has been specified, then the plot will be performed only if there is sufficient room.	
:R	Specifies that arcs and circles are to be drawn as smooth as possible.	
:P paper size	Size of paper the drawing is to be plotted on. The paper size has these values:	

ANSI Standard	DIN	Approximate Size (in.)
AH	A4H	11x8.5
AV	A4V	8.5x11
A	A4	8.5x11
B	A3	11x16
C	A2	16x22
D	A1	22x34
E	A0	34x44

Note: :P AH and :P A will produce plots of the same orientation. However, they will be located slightly different on the paper.

Two points can also be specified (for example, `:P22,30` would specify that the paper is 22 user units along the x-axis and 30 user units along the y-axis).

:T plotter type	Specifies the type of plotter when plotting to an SRM file (See the following *Plotting to a File* section).	
:A	Rotates the paper 90 degrees from the normal plotting axis. The axis rotate switch only works for the 75XX series plotters.	
file specification	The name of the output file to send the HPGL plotter commands.	must end in .ASC

Specifying the Plot Method.

You can send a drawing to the plotter by one of four methods:

PLOT ;

This method is basically a screen dump. The lower left corner of the drawing screen is plotted to the P1 corner of the paper and the rest of the screen area is proportionally plotted on the paper. This method maintains the x and y axes aspect ratios.

PLOT :S

This method uses :S to scale the plot. The lower left corner of the drawing is plotted to the P1 corner of the paper and everything on the drawing is plotted to scale from the P1 corner. This method will also include any drawing parts not currently displayed on the screen. However, if the specified scale is too large for the paper, part of the drawing may be clipped from the plot. A :S1 scale would produce a plot the same size as the drawing. (for example, a drawing five user units long would plot the same size on paper.)

PLOT :C

This method uses :C to guarantee that everything on the drawing is plotted on the paper. The system will alter the scale in order to fit the drawing on the paper.

PLOT :C :S

This method combines :C and :S to insure that the complete drawing appears to scale on the plot. If the specified scale is too large for the given paper size, the system will tell you the largest possible scale.

PLOT SPOOLING TO THE SRM

When you spool plots to a plotter connected to an SRM system, HP EGS is not able to identify information about the plotter. You must supply that information to the PLOT command which will build a data file.

Remember these points when you enter a PLOT command for an SRM plotter:

- HP EGS must be told what type of plotter you are using (different models of HP plotters behave in slightly different ways).
- HP EGS must be told what the paper size is.
- HP EGS must be told the SRM spoolfile name.
- HP EGS does not know and cannot find out where P1 and P2 are; all drawings spooled to a plotter are centered on the paper.

Use the :T and :P option to supply the information that HP EGS needs to plot a drawing when the plotter is on an SRM.

The :T option tells HP EGS the model number of the plotter connected to the SRM (for example, :T7550A). The model number must include the revision letter, in this example it is A. The following plotters can be used on the SRM:

```
7550A                   9872B
7580A/Opt. 002          9872C
7580B                   9872S
7585A/Opt. 002          9872T
7585B
7586B
```

The :P option tells HP EGS the paper size you are using. Use this command by entering a paper size defined in the message file gedit_ms.ASC or by entering the dimensions of the plot.

The message file, gedit_ms.ASC, includes definitions for all ANSI and DIN paper sizes. The definitions are broken into two groups: default paper sizes and paper sizes associated with specific plotters. The following list includes examples from gedit_ms.ASC.

Examples of default definitions for ANSI paper sizes:

```
(message tag)      (dimension values)
                     (width  height)

paper_size_A         ^D^9976 6476
paper_size_AH        ^D^9976 6476
paper_size_AV        ^D^6476 9976
paper_size_B         ^D^15112 9976
paper_size_C         ^D^20192 16072
paper_size_D         ^D^32384 21152
paper_size_E         ^D^42544 33344
```

Examples of plotter-linked definitions for ANSI paper sizes:

```
paper_size_7550_A    ^D^10170  7840
paper_size_7550_AH   ^D^10170  7840
paper_size_7550_AV   ^D^ 7840 10170
paper_size_7550_B    ^D^16450 10170
paper_size_7585_A    ^D^9976 6476
paper_size_7585_AH   ^D^9976 6476
paper_size_7585_AV   ^D^6476 9976
paper_size_7585_B    ^D^15112 9976
paper_size_7585_C    ^D^20192 16072
paper_size_7585_D    ^D^32384 21152
paper_size_7585_E    ^D^42544 33344
paper_size_9872_A    ^D^10200  8000
paper_size_9872_AH   ^D^10200  8000
paper_size_9872_AV   ^D^ 7750 10600
paper_size_9872_B    ^D^16000 11400
```

The values shown on the right are plotter units (refer to your plotter manuals for more information).

The first group shown in the gedit_ms example defines the default values for paper sizes which are not associated with specific plotters. The default values equal the smallest plottable area of any Hewlett-Packard plotter for a specific ANSI or DIN paper size.

In the second group, paper sizes are linked to specific plotters and special values. The special values usually increase the plotting area to include as much of the usable plotting surface as possible.

To add paper size definitions, or to change the values which are given here, edit the message file "gedit_ms.ASC". After you edit the message file, you must run the message file utilities which verify, test, and build an index file. Refer to Understanding HP EGS for more information on message files.

Follow these rules to define new paper size values for the message file:

- You must define a default paper size value that is not specific to any plotter model.
- You may define a paper size value that is specific to a plotter model.
- You must include values for width and height (in that order).
- You must not include the plotter model revision letter in the message file definition (for example, use _7550_ instead of _7550A_)

In the following example, two new plotter size types, BILL and JIM, have been defined, as well as two special plotter size types which are tied to the 7550 plotter. If you give a PLOT command with a :T7550A option, HP EGS uses the specially defined values, but if it is any other plotter (for example, :T7585A), HP EGS uses the default values.

```
paper_size_JIM          ^D^10000 10000
paper_size_BILL         ^D^20000 20000

paper_size_7550_JIM   ^D^12000 12000
paper_size_7550_BILL  ^D^22000 22000
```

The paper size can also be defined for HP EGS by entering the width and height (in user units) after the :P option has been entered. For example, if your drawing is in millimetres you might use the following command:

```
PLOT :C :T9872B :P400,270 #5:/PLOTSPOOL/bill.ASC
```

The final information needed to do plot spooling is the name of the spoolfile. This name must have a ".ASC" at the end of it. Your spoolfile name should be similar to this:

```
#5:/SP7550/fun.ASC
#5:/SP7586/bill.ASC
```

An SRM plot-spooling directory can be set up to allow drawings to accumulate. Drawings can then be released to the plotter one at a time by entering a message string on the SRM console. The drawings are released from the file in a first-in, first-out order. Ask your SRM system manager for the details about your SRM system.

Complete PLOT commands should be similar to this:

```
PLOT  :C  :S1 :T7550A  :PB  #5:/SP7550/bill,ASC;
PLOT  :S2  :T7586B  :PD  #5:/SP7586/Junk,ASC;
PLOT  :C  :T7550A  :PB  #5:/SP7550/fun,ASC;
PLOT  :T9872C  :PB  #5:/SP9872/bill,ASC;
```

When plotting to a file, the default is D sized paper.

Example

```
PLO :S,5 :V16 ;
```

This example plots the currently displayed drawing to the plotter. The drawing is plotted to .5 times full scale, as defined by the user units of the drawing. If the user units were inches, a four inch drawing would plot out as two inches on paper. The plot velocity is 16 centimetres per second.

Example

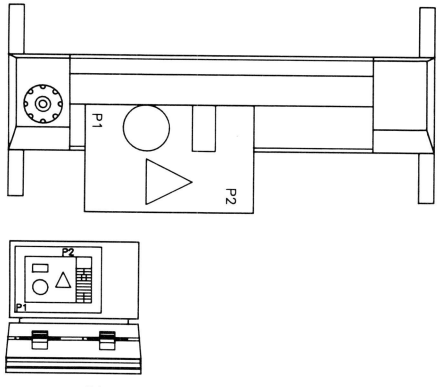
```
PLO :C :S1 :T 7580A :R :W11,8 :P AH #5:/PLOTSPOOL/AL,ASC ;
```

This example places a drawing in the SRM directory `PLOTSPOOL`. The file is `AL,ASC`. `:C` specifies that the drawing will be centered on the paper. `:S1` plots the drawing at a magnification factor of 1. The system will warn you if the magnification factor is too large and will give you the largest possible value. `:R` specifies that any arcs and circles will be plotted out as smooth as possible. `:W11,8` sets the pen width to 11.8 mils or .3 mm. `:T 7580A` specifies the plotter type. The plotter is loaded with AH paper (11 x 8.5 inches).

After you enter the command above, you must release the drawing from the SRM directory by typing in the designated message string on the SRM console keyboard. This string might be:

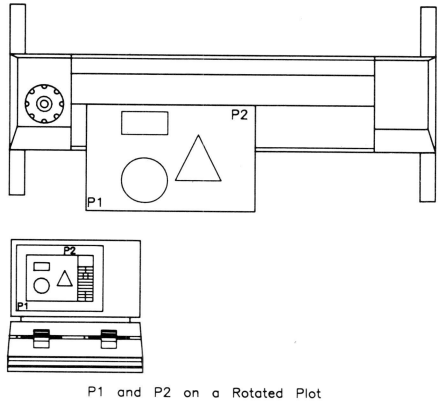
```
SP CONT 7,8 (Return)
```

(The SRM message string depends on the address of the plotter.) The screen area would plot as below:



P1 and P2 on a Standard Plot

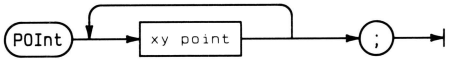
If you pressed (ENTER) and **Rotate** on the HP 7580 plotter before entering the SRM message string, the drawing would be rotated as below:



P1 and P2 on a Rotated Plot

POINT

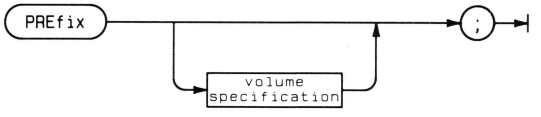
POINT displays the xy coordinates (in user units) of a specified point. This is done by entering `POINT` and selecting the point for which you wish to know the xy coordinates. This information is erased when the screen is redrawn. POINT is a secondary command.



Item	Description/Default	Range Restrictions
Parameter		
xy point	Location where you wish to find the xy coordinates.	

PREFIX

PREFIX changes the default volume to the one specified. The default volume is the first volume the the system searches when no volume name is specified. You can also prefix to a flexible disc drive. PREFIX is a secondary command.



Item	Description/Default	Range Restrictions
Parameter		
volume specification	Name of the desired default volume.	

Example

```
PRE #5:/HP_EGS/EWSYS  ;
```

This example sets the SRM volume #5: as the default unit and EWSYS as the working directory. Now you can specify a file by entering only:

```
hpegs_cn.ASC
```

If you prefixed to another volume which is not on the same unit, the working directory still is EWSYS. To specify the file you now need to enter:

```
#5:hpegs or EWSYS:hpegs
```

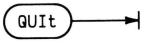
Example

```
PRE ;
```

This example displays the current prefix.

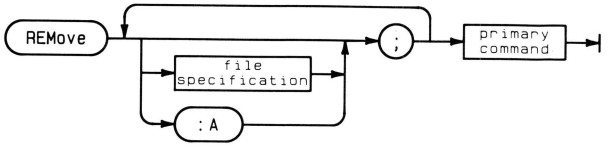
QUIT

QUIT exits the Graphics Editor. If there are any drawings which you have not saved before you enter QUIT, the system will first advise you that the drawings have not been saved. It will then ask you if you really want to quit. If you answer Y for yes, any drawings which have not been saved will be lost. If you answer N for no, the QUIT command will be ignored. QUIT is a primary command.



REMOVE

REMOVE removes a file from disc and memory. REMOVE can remove all files that are currently in memory. REMOVE is a primary command.



Item	Description/Default	Range Restrictions
Parameter		
file specification	Full path name of the file you wish to remove.	
Option		
:A	Removes all drawing files from memory. This removal does not apply to the drawing currently on the screen, nor to any drawings with instances of them in the current drawing.	

NOTE

If a drawing file exists in computer memory and on a disc, the Graphics Editor prompts you twice, once for disc removal and once for memory removal. When :A is specified, the Graphics Editor prompts you once to confirm that all drawings are to be removed from memory.

Hints

HP EGS does not allow the use of wildcards (such as $,=,?) with REMOVE.

When removing files, you must include the suffix, if applicable. The following suffixes are used by HP EGS:

`_pr.TEXT`	Graphics Editor process file
`_mc.TEXT`	Graphics Editor macro file
`_mn.TEXT`	Graphics Editor screen menu file
`_tm.TEXT`	Graphics Editor tablet menu file
`_st.TEXT`	Graphics Editor stroke table
`_cn.ASC`	Control file that defines menus for the system manager
`_ms.ASC`	Message file
`_mi.ASC`	Index to a message file (for quick loading)
`_vs.TEXT`	Volume search table
`strt.TEXT`	Start file for a module
`_d`	Drawing file
`_i`	Macro instance
`_c`	Connection list output (from Connection Lister program)
`_e`	List of errors from the connection listing
`_r`	Rat's nest output (from Generate Rat's Nest program)
`_b`	Back annotation file (from Compare Connection Lists program)
`_m`	Material list output (from LIST MAT command)
`_a`	ARCHIVE output
`_g`	GENERATE output
`_ec.TEXT`	ECHO file output
`_w`	EDIF Translator warning message file

Example

```
REM #5:/EWSYS/fun_d;
```

This example removes the drawing file `fun_d` from the SRM directory #5:/EWSYS.

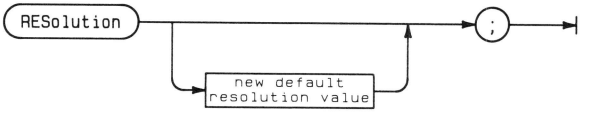
Example

```
REM EWDEMO:fun_pr.TEXT;
```

This example removes the process file `fun_pr.TEXT` from the volume `EWDEMO`.

RESOLUTION

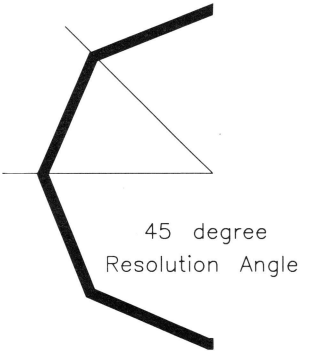
RESOLUTION sets the curve resolution of arcs, circles and ovals. This becomes the default value. Saving the process file saves this default. RESOLUTION is a secondary command.

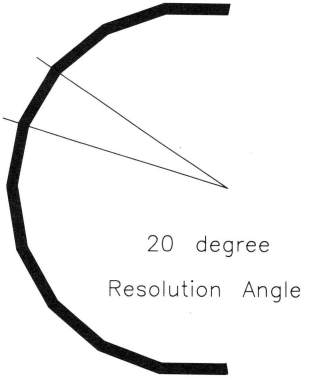


Item	Description/Default	Range Restrictions
Parameter		
new default resolution value	Desired resolution for displaying and plotting arcs and circles.	0°-120°

The Resolution Angle

The following illustrations show two arcs with different resolution angles.





The smaller the resolution angle, the smaller the line segments which approximate the circle or arc. The number of line segments equals 360 divided by the resolution angle. A 1° resolution angle would produce a circle with 360 segments ($360/1^0$). A 0° resolution will draw an arc, oval, or circle that is a smooth curve.

Although you may specify a 0° resolution, the actual resolution calculated by the system to draw a smooth curve varies with the size of a component. A small component will only require a few points and so the system will choose a resolution which makes the component appear as a smooth curve. A 0° resolution also saves memory although the system takes longer to draw the component.

The PLOT command has the option :R which overrides all resolutions and plots smooth curves. With this option, a component drawn with a coarse resolution will appear slightly different on the plot.

Example

```
RES 10  ;
```

This example changes the process file default resolution of arcs and circles to 10°. .

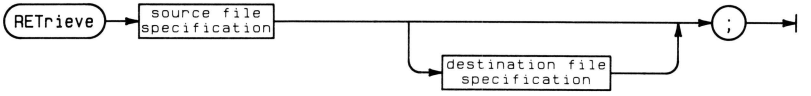
Example

```
RES ;
```

This example displays the current resolution.

RETRIEVE

RETRIEVE reconstructs an archived drawing file and any linked drawings. As the drawing is created, the individual commands are entered from the archive file and displayed on the CRT. The reconstructed drawing is saved on the specified volume. RETRIEVE is a primary command.



Item	Description/Default	Range Restrictions
Parameters		
source file specification	Specifies archive or generate file to be loaded. You must specify the suffix _a or _$.	defaults to _a
destination file specification	Specifies volume the retrieved file is to be placed on.	

Stopping the Retrieve

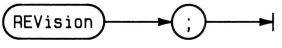
Press (STOP) to stop the retrieve process. Once the process is stopped, you are left with the partially reconstructed drawing displayed on the CRT. You may then modify or save the drawing while in the Graphics Editor. The retrieve process may be reactivated by entering the INPUT command.

NOTE

Entering the RETRIEVE command may cause the loss of some existing drawing files. This is because the RETRIEVE of an archive file begins by removing any files on the destination volume or in memory which have the same name as the archived drawing or instances within the drawing.

REVISION

REVISION displays the revision number of the HP EGS Graphics Editor. When reporting software problems to your support engineer, please supply the current revision number. REVISION is a secondary command.



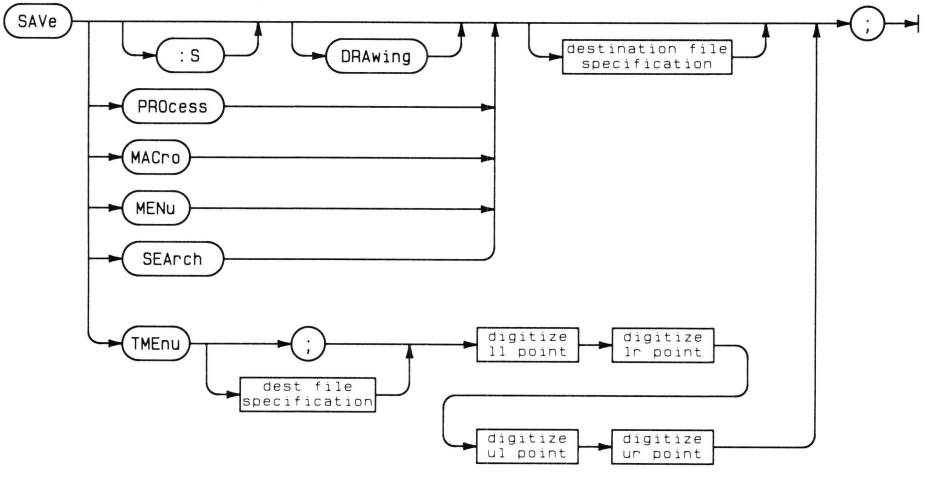
Example

```
REV  ;
```

This example displays the revision number of the Graphics Editor.

SAVE

SAVE saves instances, drawings, and files. SAVE also stores the drawing currently being edited and any named drawing (instance) linked to that drawing. SAVE is a primary command.



Item	Description/Default	Range Restrictions
Options		
:S	Specifies to smash all unnamed instances when saving.	
DRAwing	Specifies to save a drawing file. After saving, the file will have the suffix _d.	
PROcess	Specifies to save a process file. After saving, the file will have the suffix _pr.TEXT.	
MACro	Specifies to save a macro file. After saving, the file will have the suffix _mc.TEXT.	
MENu	Specifies to save a menu file. After saving, the file will have the suffix _mn.TEXT.	
SEArch	Specifies to save a volume search file. After saving, the file will have the suffix _vs.TEXT.	
TMEnu	Specifies to save a tablet menu file. Once you have drawn a tablet menu on the screen, save it like an ordinary drawing. (The tag value of 1010 should always be added to text associated to a tablet menu addition.) Then plot the drawing out on paper and place the paper copy on the graphic input device. To digitize the tablet menu, enter SAV TMENU ; and the system will ask you to digitize the four corners of the tablet menu. After saving, the file will have the suffix _tm.TEXT.	
Parameters		
destination file specification	Name of file to be saved.	8 characters maximum for drawings, 6 characters maximum for all other files in the system
digitize ll point	lower left corner of the graphics tablet.	
digitize lr point	lower right corner of the graphics tablet .	
digitize ul point	upper left corner of the graphics tablet	
digitize ur point	upper right corner of the graphics tablet.	

Saving Drawings and Instances

If you enter `SAV ;`, the system will do one of the following:

- If no drawing name is supplied with the SAVE command, then the system uses the drawing name entered with the EDIT command as the drawing file name.
- If you are within an instance edit and no name was supplied with the EDIT command, then the instance will be unnamed.
- If you are *not* within an instance edit and no name was supplied with the EDIT command, then the system displays an error message. You must then reenter SAVE with a drawing name.

If the name of the drawing being saved exists on the disc, then the user is prompted to either overwrite the existing drawing or abort the SAVE command. The system only gives this prompt if the drawing has been changed from the disc version.

You cannot enter a name for the drawing that could be interpreted by the system to be a command name or a macro name.

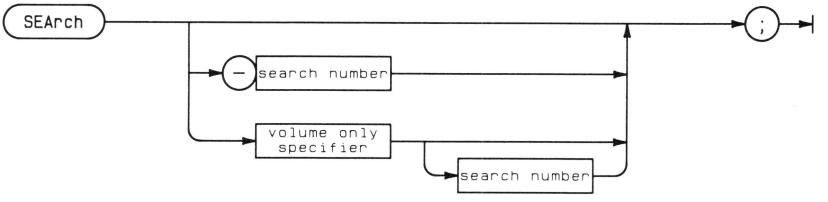
Example:

```
SAV  EWSYS:nand2 ;
```

This example saves the current drawing as `nand2` on the system volume `EWSYS`.

SEARCH

SEARCH specifies which volumes are to be contained in a search file. The volumes in this file are searched (in the order that they are listed) when you do not give a volume specification with a file name. The prefix volume begins the list. SEARCH is a secondary command.



Item	Description/Default	Range Restrictions
Option		
–	Specifies to remove the volume from the search file.	
Parameters		
search number	Order in which volume is to be placed in the search file.	0-32767
volume only specifier	Name of volume to be placed in search file.	

Example

```
SEA  ;
```

This example lists the current search file. It might read:

```
1   - ,,/EWSYS/
2   - ,,/EWEE/
3   - ,,/EWME/
4   - ,,/EWOPT/
5   - #3:
6   - #4:
```

Example

```
SEARCH  #11: 2  ;
```

This example adds volume `#11:` to the search file. It is placed after the second volume, `EWEE`. The example search file would list as:

```
1   - ,,/EWSYS/
2   - ,,/EWEE/
3   - #11:
4   - ,,/EWME/
5   - ,,/EWOPT/
6   - #3:
7   - #4:
```

Example

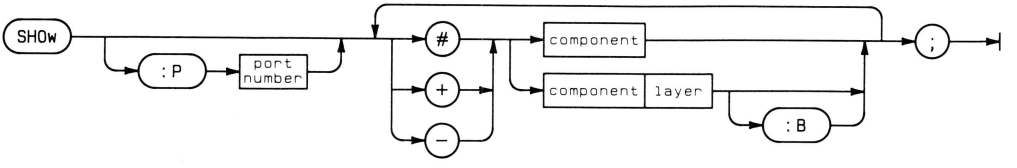
```
SEARCH   -2   ;
```

This example removes volume `EWEE` from the search file. The search file would list as:

```
1   - ,,/EWSYS/
2   - #11:
3   - ,,/EWME/
4   - ,,/EWOPT/
5   - #3:
6   - #4:
```

SHOW

SHOW specifies which layers are to be displayed, which components are to be displayed on the layers, and which components are modifiable. The effects of successive SHOW commands are cumulative. SHOW is a secondary command.



Item	Description/Default	Range Restrictions
Commands		
#	Specifies to display and allow modification of the components on the specified layer.	
+	Specifies to display and prohibit modification of the components on the specified layer.	
-	Specifies *not* to display or allow modification of the components on the specified layer.	
Options		
:P port number	Specifies which port the SHOW command operates on.	1-2
:B	Specifies that all layers bound to the affected layer are also affected.	
Parameters		
component	One of the 13 defined descriptors.	A,C,D,E,H,I,L,M,N, O,P,R,T (See the *Components* section for descriptor names.)
layer	Specifies to limit the action of the SHOW command to the layer specified. The default is all layers.	1-255

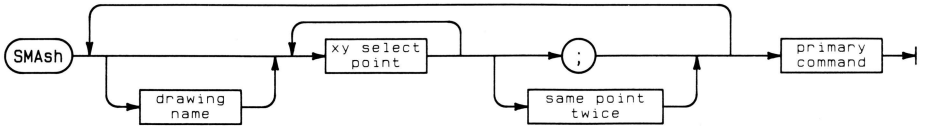
Example

```
SHO  #E -T -R2 ;
```

This example displays and allows modification of all components on all layers. However, `-T` turns off text in all layers and `-R2` turns off all rectangle components in layer 2.

SMASH

SMASH unwraps instances. The components within these instances are then displayed on the drawing and can be modified individually. Text associated to a smashed component becomes disassociated. SMASH is a primary command.



Item	Description/Default	Range Restrictions
Command		
xy select point	XY point lying near the instance to be smashed. Entering additional selection points allows you to reselect the drawing to be smashed.	
Parameter		
drawing name	Specifies the instance to smash.	

Smashing Implictly Stepped Components

If you smash an implicitly stepped instance, the components within the instance inherit the step attributes of the smashed instance. With these attributes, the components are stored in memory as implicitly stepped components and become visible in place of the smashed instances. However, if the components within an implictly stepped instance are already implicitly stepped, they cannot inherit the step attributes of the smashed instance. Therefore, the system must store all of the components as copies in order to display them in place of the smashed instances.

Because an implicitly stepped component occupies less memory than a copied component, you may wish to add the components within an instance without using implicit step. Then if you implicitly step the instance and smash it, the contained components will inherit the step attributes (instead of being copied) and occupy much less memory space.

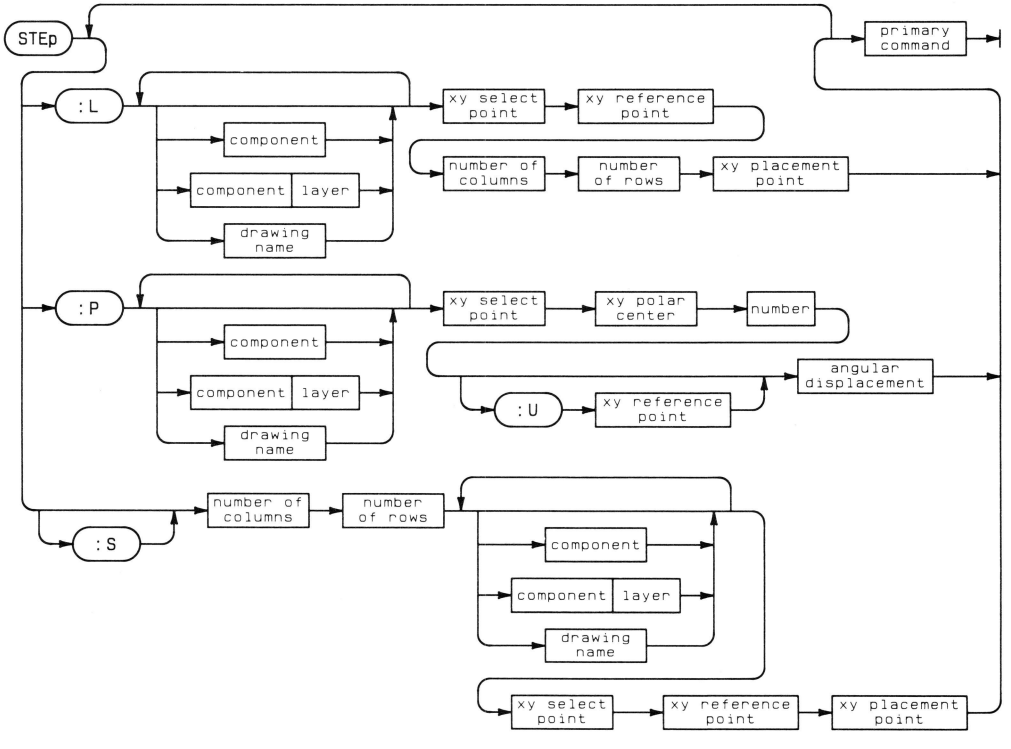
Example

```
SMA nand2 35,-70 ;
```

This example smashes the instance `nand2`. It is located at (35,-70).

STEP

STEP creates an array or radial matrix of copies of a component in the display. The STEP command is similar to the COPY command except that more than one copy can be made. Only the amount of memory in your machine limits the number of copies that can be created with the STEP command. STEP is a primary command.

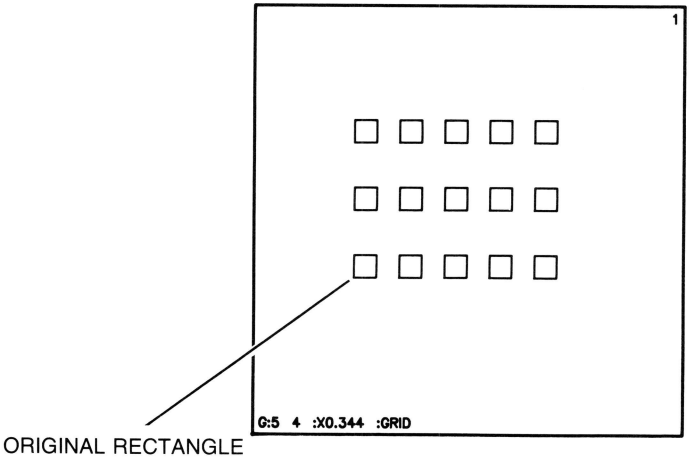


Item	Description/Default	Range Restrictions
Command		
:L	Specifies that copies are to be generated in an array-like manner.	
:P	Specifies that the copies are to be generated in a polar (radial) manner.	
Options		
:U	Specifies that each polar copy is NOT to be rotated as the copies are generated.	
:S	This switch is leftover to remain compatable with previous EGS versions. It does nothing.	
Parameters		
component	One of the 13 defined descriptors. The default descriptor is E.	A,C,D,E,H,I,L,M,N, O,P,R,T (See the *Components* section for descriptor names.)
layer	Layer containing the component to be copied.	1-255
drawing name	Name of a specific instance component to be copied.	
xy select point	XY location lying on or inside the component you wish to copy.	
xy reference point	XY location which acts as a reference point for copying.	
number of columns	Number of columns in the matrix of copies of the selected component.	
number of rows	Number of rows in the matrix of copies of the selected component.	
xy placement point	XY location which specifies the X and Y distances between the reference points of the components. This defines the distance between the origin of each component in the matrix.	
xy polar center	The point about which the polar copies are to be generated.	
number	The number of polar copies.	
xy reference point	The point on the component about which the rotation is to take place.	
angular displacement	The amount (in degrees) between each polar copy	0° - 360°

Example

```
STE  :L R 10,10 0,0 5 3 20,30 ;
```

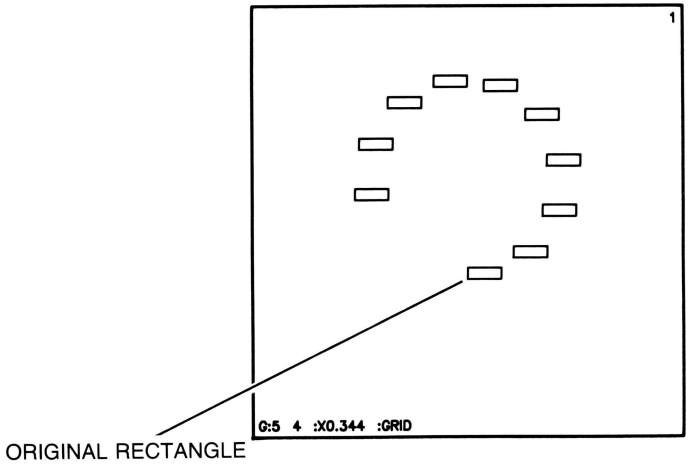
This example makes fifteen copies of a rectangle located at (10,10). An origin of (0,0) is assigned to the original rectangle. The copies are arranged in a 5 (column) by 3 (row) matrix. Since the xy placement point is (20,30), the copies are spaced with their origins separated by 20 units in the x direction and 30 units in the y direction.



Example

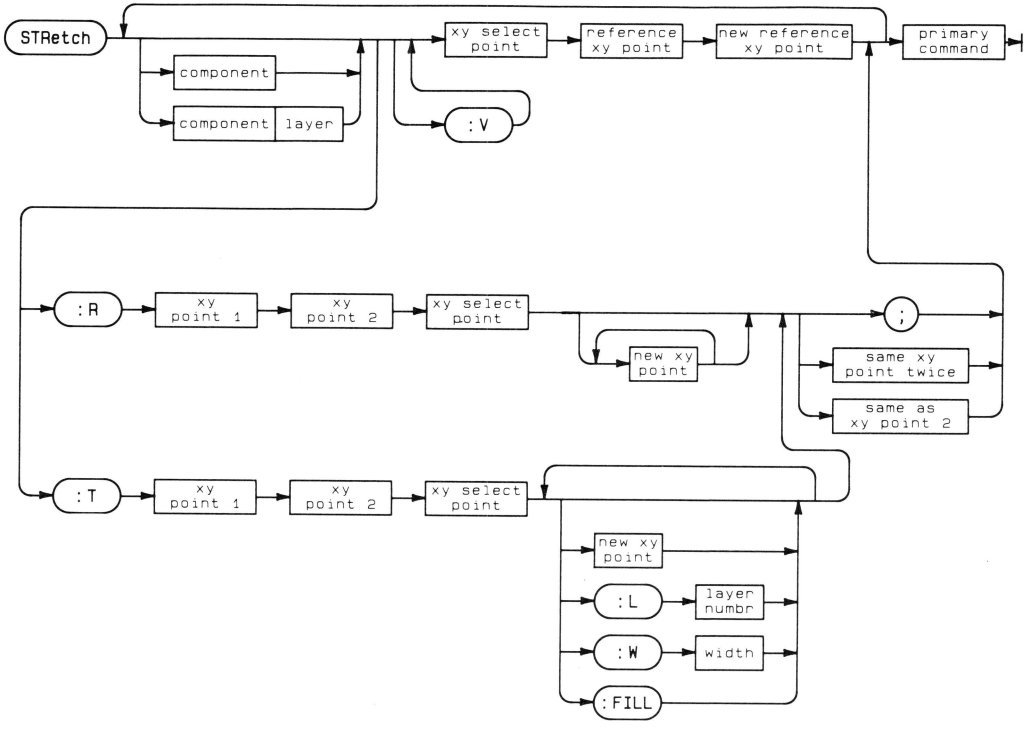
```
ADD R 0,0 30,10;
STE  :P R  0,0 0,90 9 :U 15,5   30 ;
```

This example first adds a rectangle. The second command generates nine radial copies about the xy polar center (0,90). The xy select point on the rectangle is (0,0). The copies are not rotated since `:U` is specified. The xy reference point (15,5) lies on the original rectangle. On each copy this point is the same distance from the xy polar center. The copies are separated by 30°.



STRETCH

STRETCH stretches or replaces a side, edge, or vertex of the following components: arc, circle, dimension, line, rectangle, or polygon. STRETCH is a primary command.



Item	Description/Default	Range Restrictions
Command		
xy select point	XY location lying on or inside the component you wish to stretch or the portion of the component that you want to replace.	
reference xy point	XY point identifying the part of the component to be stretched.	
new ref xy point	XY location which specifies the new location for the point being stretched.	
Options		
:V	Specifies that a vertex is to be added to a line segment, rectangle or polygon.	
:R	Specifies to replace a portion of a polygon, rectangle or line segment component with zero or more vertices. If neither of the xy points is a vertex, the component will rubber band when stretched. If one of the xy points is a vertex, the vertex will rubber band to the new xy point. When working with polygons and line endpoints, the :R option cannot be terminated by reselecting xy point 2. Any component that has been modified with STRETCH :R is no longer part of any collection (see GATHER).	
:T	Specifies to replace a portion of a line segment with zero or more vertices. :T only applies to lines and will preserve connectivity of lines extending across layers. Unlike :R, you cannot relocate line endpoints or extend a line with :T.	
:L	Specifies to begin a new line at the layer indicated.	
:W	Specifies to begin a new line with the width indicated.	
:FILL	Specifies to begin a new line which is filled.	

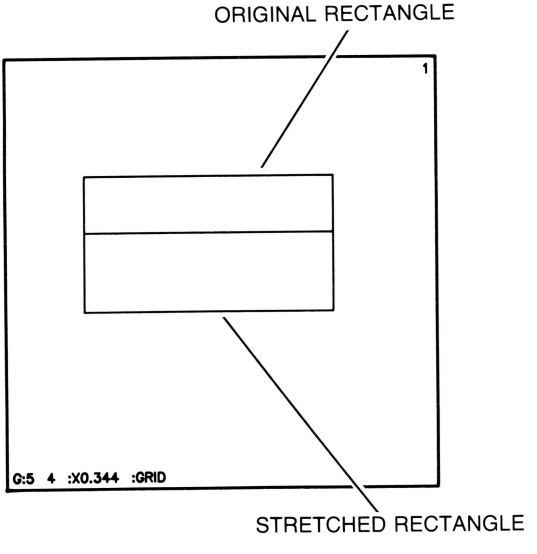
(Table continued)

Item	Description/Default	Range Restrictions
Parameters		
component	Descriptor specifying an arc, circle, dimension, line, rectangle, or polygon to stretch	A,C,D,L,P,R
layer	Layer containing the component to be stretched.	1-255
xy select pt.	Point identifying the component to stretch.	
reference xy point	Point where you wish to stretch the component.	
new reference xy point	Point to where you wish to stretch the component.	
xy point 1	A starting point for the replacement set of vertices.	
xy point 2	An ending point for the replacment vertices.	
new xy point	Subsequent xy locations for specifying the portion of the component being replaced or for identifying new replacement vertices.	
layer number	Layer in which to place the new line.	1-255
width	Width for the new line in current user units.	0 through the width of the screen area.

Example

```
ADD R -70,100 150,150;
STR -70,100  80,100 40,30 ;
```

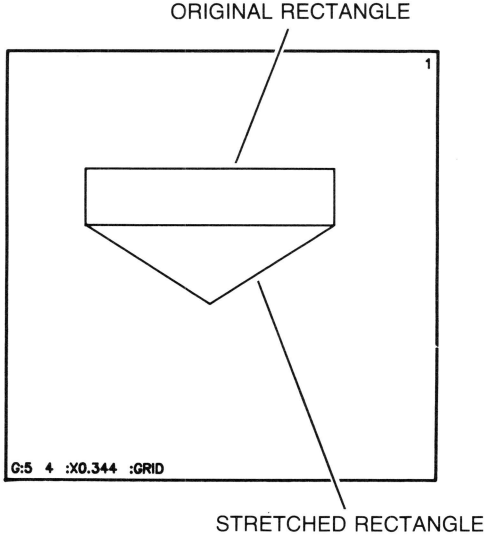
This example relocates the rectangle so that the reference xy point (80,100) is now located at (40,30).



Example

```
ADD R -70,100 150,150;
STR :V -70,100 80,100 40,30;
```

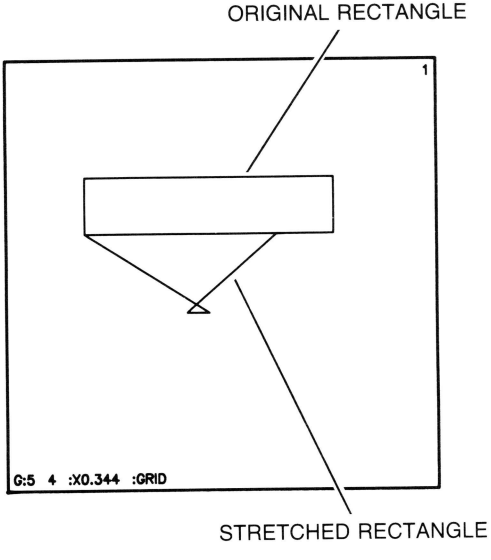
This example adds a vertex and stretches it to (40,30). Adding a vertex to a rectangle converts the rectangle to a polygon.



Example

```
ADD R -70,100 150,150;
STR :R -70,100 100,100 75,100 40,30 20,30;
```

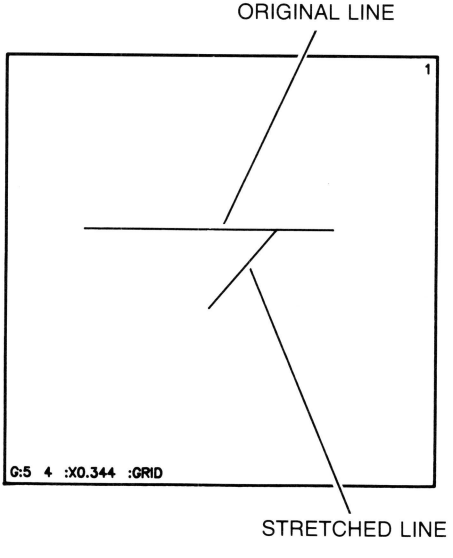
This example replaces part of the side of the rectangle with two more vertices at (40,30) and (20,30). The portion of the rectangle between (-70,100) and (100,100) is deleted. The rectangle is converted to a polygon.



Example

```
ADD L -70,100 150,100;
STR :R -70,100 100,100 75,100 40,30;
```

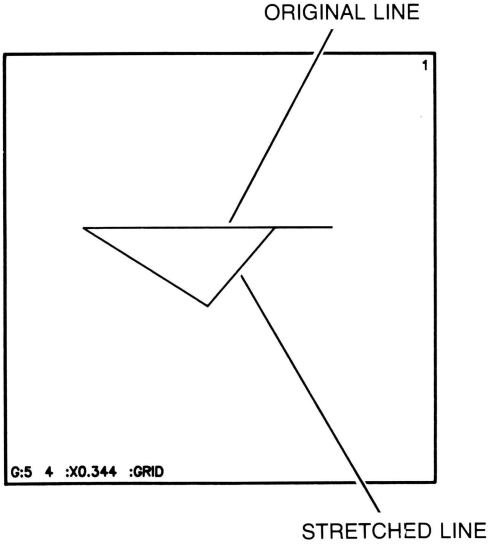
This example deletes the line from (-70,100) to (100,100). That portion of the line is replaced with a line extending from (100,100) to (40,30).



Example

```
ADD L3 -70,100 150,100;
STR :T -70,100 100,100 75,100 40,30 ;
```

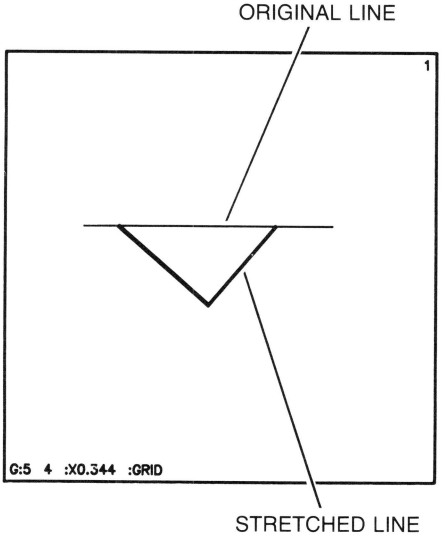
This example uses `:T` to stretch the line. Notice that the line is stretched from the vertex and connectivity is maintained. This is unlike the previous example which used `:R` to relocate the vertex without maintaining connectivity.



Example

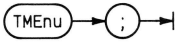
```
ADD L3 -70,100 150,100;
STR :T -40,100 100,100 75,100 :W4 :L6 :FILL 40,30 ;
```

This example uses `:T` to stretch the line. The stretched portion has a width of 4, is placed on layer 6 and filled.



TMENU

TMENU toggles the tablet menu on and off. TMENU is a secondary command.



Use of TMENU

When the tablet menu is toggled on, the cursor will appear on the screen area only when the stylus is on the section of the tablet defined as the drawing area. If the tablet menu is toggled off, the cursor will appear on the screen area any time the stylus is on the tablet.

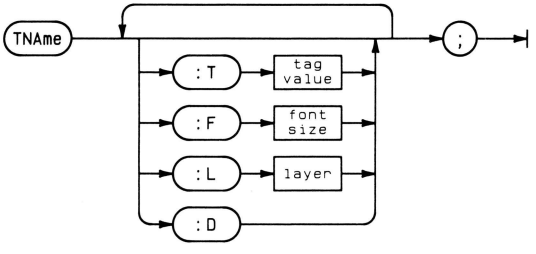
The system turns off the tablet menu when a mouse is used as a locator device.

To save a tablet menu, see the SAVE command.

To load a tablet menu, see the LOAD command.

TNAME

TNAME defines the tag , font size, layer, and display option values of TNAME associated text. TNAME is used when transferring archived EGS-45 files to HP EGS since HP EGS associated text has some parameters which are not part of the EGS-45 archive file. The default values are stored in the process file and may be changed at any time. TNAME is a secondary command.



Item	Description/Default	Range Restrictions
Options		
:T tag value	The default tag value for TNAMEs entered from EGS-45 archive files.	
:F font size	The default font size for TNAMEs entered from EGS-45 archive files.	
:L layer	Sets the default layer for TNAMEs entered from EGS-45.	1-255
:D	Specifies that TNAMEs entered from EGS-45 archive files will be displayed.	

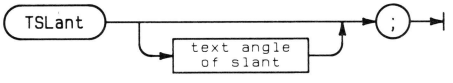
Example

```
TNA :T 1005 :F 10 :D :L 5 ;
```

This example defines the tag value as 1005, the font size as 10, and the layer as 5 for an archived EGS-45 file transferred to HP EGS.

TSLANT

TSLANT sets the text slant. This becomes the process file default value. Saving the process file saves this default. TSLANT is a secondary command.

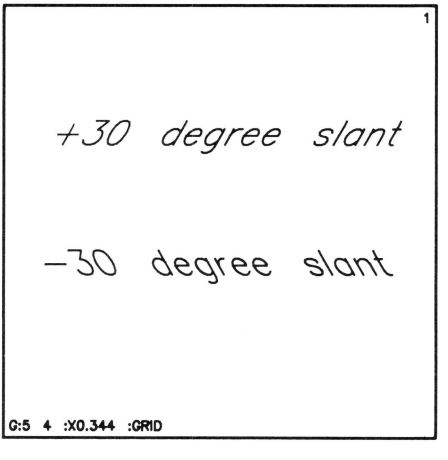


Item	Description/Default	Range Restrictions
Parameter		
text slant	Specifies the angle of text slant measured from an imaginary vertical line. A positive angle is measured clockwise. The default is 0°.	-75° - 75°

Examples

```
TSL 30 ;
```

This example sets the slant of notes and text 30° to the right.

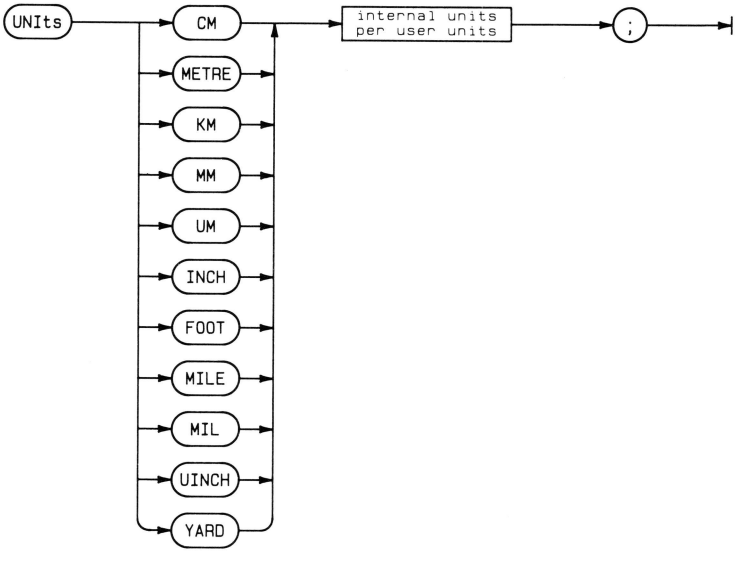


UNDO

UNDO reverses the effects of the command immediately preceding it. UNDO affects: ADD, COPY, DELETE, GATHER, GROUP, MODIFY, MOVE, SMASH, STEP, STRETCH,and WRAP. UNDO is a primary command.

UNITS

UNITS assigns the physical distance between system grid points, thereby determining the drawing resolution. If you change the user units of a drawing currently in memory, the system may prompt that the new user units do not match the drawing units. You are then asked if you still wish to change the user units. Entering Y for yes causes the system to automatically convert the units of the drawing to match the new user units. UNITS is a secondary command.



Item	Description/Default	Range Restrictions
Commands		
CM	Specifies centimetres as the units.	
METRE	Specifies metres as the units.	
KM	Specifies kilometres as the units.	
MM	Specifies millimetres as the units.	
UM	Specifies micrometres as the units.	
INCH	Specifies inches as the units.	
FOOT	Specifies feet as the units.	
MILE	Specifies miles as the units.	
MIL	Specifies mils as the units.	
UINCH	Specifies micro-inches as the units.	
YARD	Specifies yards as the units.	
Parameter		
internal units per user units	Specifies the number of system grid points that are equal to one user unit.	

Example

```
UNI INCH, 100;
```

This example sets the the distance between 100 system grid points to 1 inch. Therefore each grid point is .01 inch apart. Since an xy location can only be entered on a grid point, the resolution is limited to .01 inches.

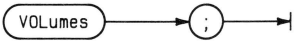
Example

```
UNI INCH 1  ;
UNI FOOT 1;
```

The first command of this example sets each system grid point one inch apart. The second command changes the spacing to one foot between each system grid point. If a drawing was made using the first units, a four inch long component would now be four feet long.

VOLUMES

VOLUMES lists the volumes currently on-line. VOLUMES is a secondary command.



Example

```
VOL;
```

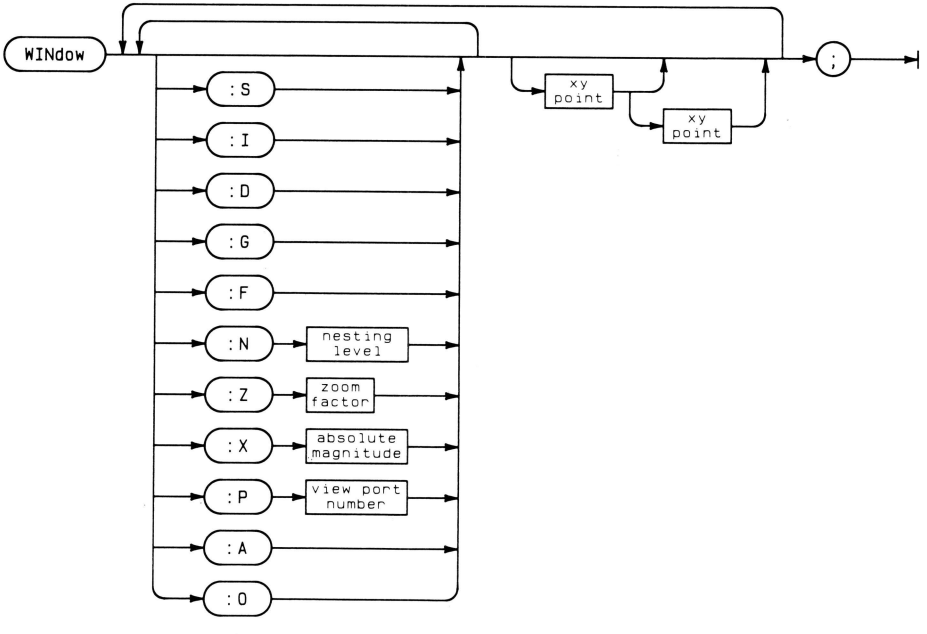
This example list the current volumes on line. The list might appear as:

```
  VOLUMES on-line:
  1   CONSOLE:
  2   SYSTERM:
  3 # MINI3:
  4 # MINI4:
  5 # MY_SRM:
  6   PRINTER:
 45 * SYSTEM04:
Prefix is - MY_SRM:
```

The number on the far left is the logical unit number associated with the volume. The * in the second column indicates the system volume. The # indicates all other block-structured volumes currently on-line. The remaining volumes (shown with no character in the second column) are non-disc. The last line of the display shows the current default volume. It is where the system looks for a file when no volume has been specified.

WINDOW

WINDOW controls the display mode, magnification and location of the window, sets the depth of the displayed nesting level and grid display, and redraws the screen. WINDOW is a secondary command.



Item	Description/Default	Range Restrictions
Options		
:S	Sets the display to the symbolic mode. In this mode, all components entered on layers defined as symbolic layers and interconnect layers are displayed.	
:I	Sets the display to the interconnect mode. In this mode, only components entered on layers defined as interconnect layers are displayed. Lines with width in interconnect layers are displayed without width.	
:D	Sets the display to the detail mode. In this mode, all components entered on layers defined as detail layers and as interconnect layers are displayed.	
:G	Toggles the grid on and off.	
:F	Specifies to redraw the screen and adjust the window such that the entire drawing fits into the drawing area of the CRT.	
:N nesting level	Specifies the nesting level depth to be displayed. The nesting depth at entry is nesting level 1.	
:Z zoom factor	Specifies the magnification of the drawing displayed on the CRT. Positive values (>0) cause the system to zoom in and enlarge the components. Negative values (<0) cause the system to move out and show a larger screen area. Consequently, the components appear smaller. If no zoom factor is supplied with the :Z parameter, the window is adjusted to the previous window setting.	
:X absolute magnification	Specifies the absolute magnification of the drawing. This specifies the relationship between user units and the magnification of the drawing in the display.	

(Table continued)

Item	Description/Default	Range Restrictions
:P viewport number	Specifies the current and activated viewport. A + or a space preceding the viewport number turns on the viewport. A - preceding the viewport number turns off the viewport. Once the viewport is turned off, it cannot be updated and components will not be highlighted. However, by entering `WIN` again, you can choose points from the frozen viewport.	
:A	Toggles the display of non-displayed associated text. The default is off.	
:O	A toggle which specifies to draw just the outlines of components (such as previously filled components will be redrawn without fill).	
Parameter		
xy point	Selects area of drawing to be shown on the CRT. If one xy point is entered, the window is shifted so this xy point becomes the center of the drawing. This is called panning and maintains the scale. If a second xy point is entered, the window is changed so the smaller x value and the smaller y value of the two points become the xy coordinates of the lower left corner of the window, and the larger x and y become the xy coordinates of the upper right corner of the window. The magnification of the drawing viewed through the window is then changed such that the entire rectangular area defined by xy points 1 and 2 is contained in the window. This is called windowing in.	

Example

```
WIN  :S :N3 :Z5 :X2 ;
```

This example sets the display to the symbolic mode, the displayed nesting depth to 3 (components in nesting levels 1, 2, and 3 are displayed) and zooms in by a factor of 5. If the user units are inches, :X2 causes a drawing 1 user unit long (1" long) to occupy 2 inches of the CRT when displayed.

Example

```
WIN :Z ;
```

This example adjusts the window zoom to the previous setting. For example, if the zoom was +3 before you changed it to -5, `WIN :Z ;` would reset the zoom to +3. Thus, by repeatedly entering `WIN :Z ;`, you can toggle between two different window settings.

Example

```
WIN ;
```

This example redraws the screen.

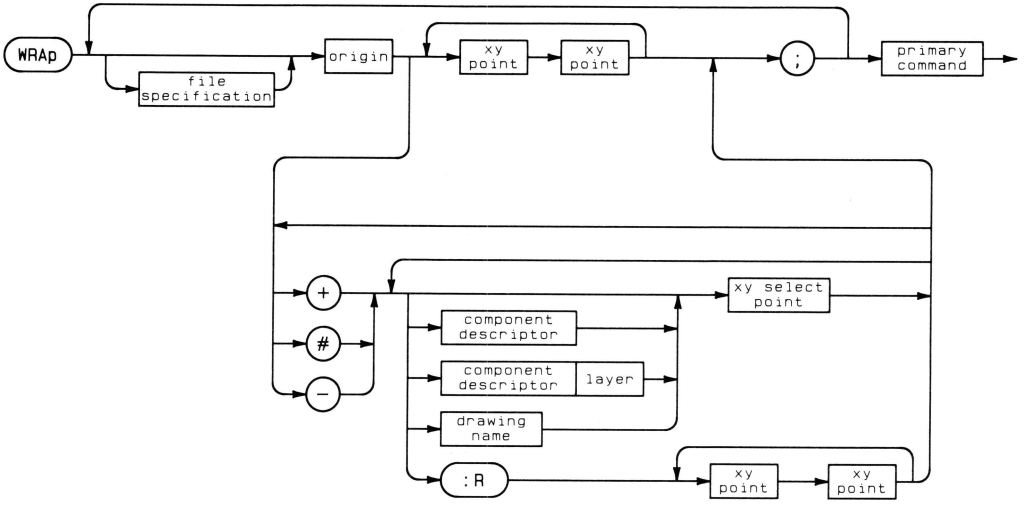
Example

```
WIN :P2 :P-1;
```

This example makes viewport 2 the current and activated port and deactivates port 1. No points can be entered in port 1 until `WIN` is re-entered.

WRAP

Wrap creates a component called an instance. The instance replaces the components contained within its boundary and these components are placed on the next higher nesting level. The instance becomes a link between the original drawing and the replaced components. Because the components contained in the instance are only linked to the original drawing, they can be modified without affecting the drawing. A copy of the instance is placed in computer memory and if named and saved, the instance can be referenced to other drawings. WRAP is a primary command.



Item	Description/Default	Range Restrictions
Command		
origin	XY location specifying the origin of the instance.	
xy points	XY locations specifying the opposite corners of the instance rectangle(s).	
Options		
+	Include the component specified by xy select point(s).	
#	Same as +.	
-	Exclude the component specified by xy select point(s).	
:R	Include or exclude the components within a rectangular area defined by two xy points.	
Parameters		
file specification	Name assigned to the instance. If named and saved, the instance becomes a library part which can be added to other drawings. You cannot enter an instance name that could be interpreted by the system as a command or macro name.	file name 8 characters maximum
component descriptor	One of 12 defined descriptors. Specifies a component to include or exclude from the instance.	A,C,D,E,H,I,L,M,N, P,R,T (See the *Components* section for descriptor names.)
layer	Layer containing the selected components.	1-255
drawing name	Name of an instance you want to wrap.	
xy select point	XY location of the component you wish to select or deselect. Entering additional xy select points allows additional components to be included or not included in the drawing you are creating.	
xy point	XY points defining a rectangular selection area. Multiple rectangular areas can be specified.	

Use of Wrap

Only components completely within the wrapping rectangle become part of the new instance. WRAP differs from GATHER in three ways:

- WRAP only wraps components. GATHER groups components *and* vertices.
- WRAP places the wrapped components on the next higher nesting level. GATHER does not.
- When moved or stretched, a wrapped instance does not maintain its connections. Connections are maintained when a gathered group is moved.

If you do not specify a name when creating an instance, it is stored as an unnamed instance. The system preserves unnamed instances as a part in a retrieved drawing but does not allow these instances to be added to other drawings.

The following rules apply when using `WRAP +` in conjunction with associated text:

- If you wrap a component with associated text, you will get the component and all associated text.
- If you wrap a text/note component which is associated to some other component (:AC), you will get the associated text and the component.
- If you attempt to wrap a text/note component which is associated with a drawing (:AD), you will not be allowed to wrap it.

Example

```
WRAP nand2 0,0    25,10   200,100  ;
```

This example creates the instance nand2 with an origin located at (0,0). The instance is composed of all components within the wrapping rectangle defined by locations (25,10) and (200,100). If you save the instance before exiting the the Graphics Editor, the instance becomes a library part.

Example

```
WRAP gate  50,50  + L 60,75  C2  90,80  I 100,120  ;
```

This example creates an instance called `gate`. It has its origin at (50,50). The instance replaces:

- A line which passes through the location (60,75). The descriptor L specifies that only a line can be selected.
- A circle on layer 2 which contains the point (90,80). The descriptor C2 specifies that only a circle on layer 2 can be selected.
- The instance located at the point (100,120).

These components are on the next deeper nesting level than the instance `gate`. The instance is now a component which is linked to the drawing.

Archive Reference

Information Provided

This Archive Reference applies to HP EGS 2.0 and later versions of the software. The reference contains two sections: a section outlining the archive file format by explaining a sample archive file, and a section alphabetically listing the HP EGS commands contained in an archive file. Each command in the second section is explained with a syntax diagram and table. For information on interpreting the syntax diagrams, see the introduction to the *HP EGS Syntax Reference*.

Archive File Format

An archive file contains a list of commands to re-create a drawing and any instances contained in that drawing. As an example, the command:

```
ARC :P bolt ;
```

archives the drawing `bolt`. The `:P` option specifies that the process file is listed at the beginning of the archive file. The archive file is stored in the prefix volume as bolt_a and contains the following commands:

```
$$ HP EGS Graphics Editor
$$ Revision      : 2.0
$$ Archival Date : 9/13/84
$$ Archival Time : 3:16 PM

UNITS INCH,10000;
LOCK 0.0000;
GRID 0.1250,2 0.0000,0.0000;
DECIMALS 3;
FSIZE 0.1250;
TSLANT 0.0000;
RESOLUTION 0.0000;
DIM_FSIZE 0.1250;
DIM_TFSIZE 0.0938;
DIM_TSLANT 0.0000;
DIM_DEC 3;
DIM_SCALE 1.000;
DIM_RADIX ".";
DIM_EXTEND 0.1250;
DIM_DIR HORIZONTAL;
LEVEL 1;
DIM_UNITS NORMAL;
INAME :L1 :F0.1000 :T1001;
TNAME :L1 :F0.1000 :T1004;
$$                Layer  Line
$$  Label         Type   Type    Color   Pen     Layer
EQU INSTBOUN     :D     :L8     :C1     :P1     0;
EQU   SOLID0     :D     :L1     :C4     :P7     1;
:
EQU  EXTRA69     :D     :L1     :C7     :P6     69;
EXIT;

(Archive file continued)
```

```
$FILES  bolt, nutsd $;

EDIT;
EDIT;

DEF :I NUTSD { "ENTer the nut diameter," BDIA } 'DEF OLDSNAPMODE
            ("'"+SYSTEM_SNAP_MODE+"'"); :RAW '"ADD A11
            (0,6*{BDIA}),(0,97*{BDIA}) (0,6*{BDIA}),(0,485*{BDIA}) ""
            (0,75*{BDIA}),(0,7275*{BDIA}) ; "" ""
            (0,6*{BDIA}),(0,485*{BDIA}) (0,6*{BDIA}),(-0,485*{BDIA}) ""
            (0,75*{BDIA}),0; ""
            (0,6*{BDIA}),(-0,485*{BDIA}) (0,6*{BDIA}),(-0,97*{BDIA}) ""
            (0,75*{BDIA}),(-0,7275*{BDIA}) ; "" L11
            (0,6*{BDIA}),(-0,97*{BDIA}) 0,(-0,97*{BDIA}) """"
            0,(0,97*{BDIA}) (0,6*{BDIA}),(0,97*{BDIA}) ; ""
            (0,75*{BDIA}),(-0,7275*{BDIA})
            (0,75*{BDIA}),(0,7275*{BDIA});"" 0, (-0,485*{BDIA})
            (0,6*{BDIA}),(-0,485*{BDIA}) ; "" 0,(0,485*{BDIA})
            (0,6*{BDIA}),(0,485*{BDIA}) ; "" OLDSNAPMODE ";

EDIT bolt;
SHOW #E;
LOCK 0,0;
LEVEL 1;
GRID 0,0001,1 0,0;
WINDOW 0,0000,-0,2425 0,1875,0,2425;
ADD nutsd ,25
:ND  0,0000,0,0000;
GRID 0,1250,2 0,0000,0,0000;
LOCK 0,0000;
SAVE;
```

Discussion of Example

All archive files have a similar format of commands. This section explains the format by outlining the commands of the previous example.

File Information

All archive files begin with four comment lines similar to the following:

```
$$ HP EGS Graphics Editor
$$ Revision      : 2,0
$$ Archival Date : 9/13/84
$$ Archival Time : 3:16 PM
```

These lines identify the file as an HP EGS file and list the archival date and time. Notice that each of these comment lines begin with `$$`. The system ignores any line in an archive file which begins with `$$`.

Process File

The process file is listed next if :P was specified during archiving.

```
UNITS INCH,10000;
LOCK 0.0000;
GRID 0.1250,2 0.0000,0.0000;
DECIMALS 3;
FSIZE 0.1250;
TSLANT 0.0000;
RESOLUTION 0.0000;
DIM_FSIZE 0.1250;
DIM_TFSIZE 0.0938;
DIM_TSLANT 0.0000;
DIM_DEC 3;
DIM_SCALE 1.000;
DIM_RADIX ".";
DIM_EXTEND 0.1250;

(Process file continued)

DIM_DIR HORIZONTAL;
LEVEL 1;
DIM_UNITS NORMAL;
INAME :L1 :F0.1000 :T1001;
TNAME :L1 :F0.1000 :T1004;
$$                Layer  Line
$$  Label         Type   Type    Color   Pen     Layer
EQU INSTBOUN     :D     :L8     :C1     :P1     0;
EQU   SOLID0     :D     :L1     :C4     :P7     1;
:
EQU EXTRA69      :D     :L1     :C7     :P6    69;
EXIT;
```

It is recommended that :P is always specified during archiving because the process file you are currently using may differ from the one used in the original drawing. If the process files are different, and :P was *not* specified during archiving, the incompatible process files may prohibit you from retrieving the drawing. You may, however, circumvent the problem of incompatible process files by modifying the current process file to match the one used when the drawing was archived.

$FILES
The command

```
$FILES  bolt, nutsd $;
```

follows the optional process file. `$FILES` is the first command of the retrieve process. Listed after `$FILES` is the main drawing name and any instances contained in the drawing. In this case, `bolt` is the main drawing name and `nutsd` is an instance contained in `bolt`.

NOTE

When retrieving an archived drawing, $FILES erases any drawings and instances in the destination volume and in memory that have the same name as the drawings and instances in its list.

If any leader lines were used on the original drawing, the instance L_arrow is included in the $FILES list. Similarly, the instance d_arrow is included if the original drawing had any dimensions.

The system retrieves the drawing and instances in the reverse order of their listing after $FILES. The order of the $FILES list is determined by the nesting level of the instance; the deepter the nesting level of the instance, the later it appears in the $FILES list. Therefore, instances on the deepest nesting level are retrieved first.

EDIT;
The commands:

```
EDIT;
EDIT;
```

clear the drawing screen in preparation for the retrieval.

Retrieval Methods

The commands following the two EDITs begin retrieval of any instances and the main drawing. As the system retrieves each instance, the retrieval method is based on the following criteria:

- If the instance is a macro instance and :M was *not* specified during archiving, the instance is retrieved with DEF :I. In the example, `nutsd` is a macro instance which was archived without :M specified. Therefore, it is retrieved with DEF :I.
- If the instance or drawing is *not* a macro instance (that is, a graphical instance) or :M *was* specified during archival, the instance or drawing is retrieved with EDIT. In the example, `bolt` is a drawing and therefore retrieved with EDIT.

Retrieval with DEF :I

The first method of retrieval uses DEF :I. This command could be considered a static command because it does not draw the instance on the screen but only textually defines it. The macro instance `nutsd` is defined as:

```
DEF :I NUTSD { "ENTer the nut diameter," BDIA } 'DEF OLDSNAPMODE
             ("'"+SYSTEM_SNAP_MODE+"'"); :RAW '"ADD A11
             (0,6*{BDIA}),(0,97*{BDIA}) (0,6*{BDIA}),(0,485*{BDIA}) ""
             (0,75*{BDIA}),(0,7275*{BDIA}) ; "" ""
             (0,6*{BDIA}),(0,485*{BDIA}) (0,6*{BDIA}),(-0,485*{BDIA}) ""
             (0,75*{BDIA}),0; ""
             (0,6*{BDIA}),(-0,485*{BDIA}) (0,6*{BDIA}),(-0,97*{BDIA}) ""
             (0,75*{BDIA}),(-0,7275*{BDIA}) ; "" L11
             (0,6*{BDIA}),(-0,97*{BDIA}) 0,(-0,97*{BDIA}) """"
             0,(0,97*{BDIA}) (0,6*{BDIA}),(0,97*{BDIA}) ; ""
             (0,75*{BDIA}),(-0,7275*{BDIA})
             (0,75*{BDIA}),(0,7275*{BDIA});"" 0, (-0,485*{BDIA})
             (0,6*{BDIA}),(-0,485*{BDIA}) ; "" 0,(0,485*{BDIA})
             (0,6*{BDIA}),(0,485*{BDIA}) ; "" OLDSNAPMODE ";
```

The system does not draw the instance on the screen until the macro instance file is added with its parameters, location, and any options. In the example, the command

```
ADD nutsd ,25 :ND 0,0000,0,0000;
```

adds the macro instance. The nut diameter (BDIA) is 0.25 and the nut is located at (0,0). `:ND` specifies that default text will not be associated with the instance.

Retrieval with EDIT

The second method of retrieval uses EDIT to re-create graphical instances, drawings, or macro instances if :M was specified during the archiving. The following sections discuss retrieval with EDIT.

If :M had been specified during archiving, the macro instance `nutsd` would be archived as an unnamed graphical instance. This unnamed instance would no longer be a macro definition but only the following list of commands:

```
EDIT NONAM000;
SHOW #E;
LOCK 0,0;
LEVEL 1;
GRID 0,0001,1 0,0;
WINDOW  0,0000,-0,2425  0,1875,0,2425;
ADD A11 :W0,0000 :R0  0,1500,0,1213  0,1500,0,2425  0,1875,0,1819;
ADD A11 :W0,0000 :R0  0,1500,-0,1213  0,1500,0,1213  0,1875,0,0000;
ADD A11 :W0,0000 :R0  0,1500,-0,2425  0,1500,-0,1213  0,1875,-0,1819;
ADD L11 :W0,0000  0,1500,-0,2425  0,0000,-0,2425  0,0000,0,2425
0,1500,0,2425;
ADD L11 :W0,0000  0,1875,-0,1819  0,1875,0,1819;
ADD L11 :W0,0000  0,0000,-0,1213  0,1500,-0,1213;
ADD L11 :W0,0000  0,0000,0,1213  0,1500,0,1213;
GRID 0,1250,2  0,0000,0,0000;
LOCK 0,0000;
EXIT;
```

NOTE: Do not begin an instance name with NONAM or the instance will become unnamed during archiving and/or saving.

Each graphical instance or drawing is retrieved with a cycle of commands that begins with `EDIT` and the instance or drawing name. The current example does not contain any graphical instances or macro instances archived with :M. Therefore the archive file contains only one EDIT cycle which begins with:

```
EDIT bolt;
```

The cycle continues with the SHOW, LOCK, LEVEL, GRID, and WINDOW commands, a series of ADD commands, and the GRID, LOCK, and SAVE or EXIT commands.

The first five commands set the system to the necessary resolution for accurate drawing retrieval.

```
SHOW #E;
LOCK 0,0;
LEVEL 1;
GRID 0,0001,1 0,0;
WINDOW  0,0000,-0,2425  0,1875,0,2425;
```

The ADD commands build the graphical instance or drawing by adding the necessary components. As previously mentioned, the command

```
ADD nutsd ,25 :ND 0,0000,0,0000;
```

adds the macro instance nutsd to the main drawing.

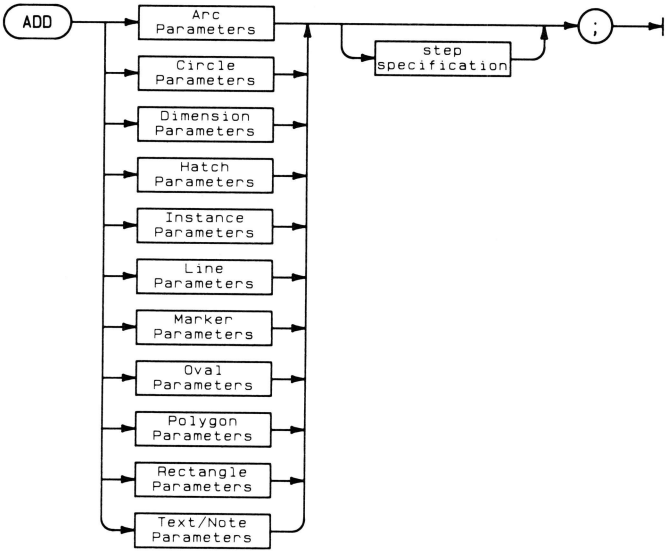
The GRID and LOCK commands reflect the most recent grid and lock settings.

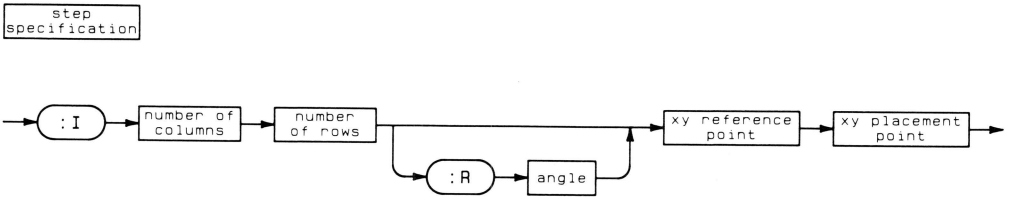
```
GRID 0,1250,2  0,0000,0,0000;
LOCK 0,0000;
```

Each graphical instance or drawing loop ends with a SAVE; , or if the instance is unnamed, EXIT; .

ADD

ADD is the primary command of archive files since it adds all components. However, to add components a descriptor must be included with the ADD command. ADD is shown here without descriptors to illustrate the implicit step specification.

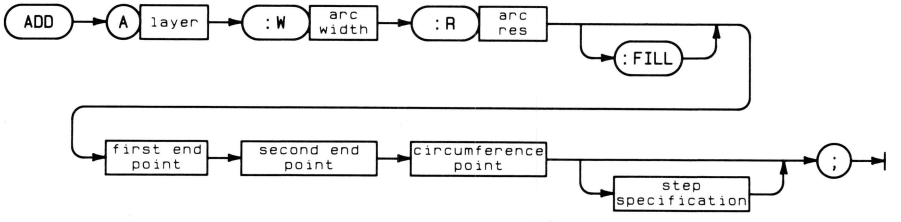




Item	Description/Default	Range Restrictions
step specification	Specifies to add an implicit step.	
number of columns	The number of columns in the array	
number of rows	The number of rows in the array	
:R	Specifies to rotate the array.	
angle	The angle of rotation of the array. This is measured from an imaginary horizontal line.	
xy reference point	The lower left corner of the component's boundary rectangle.	
xy placement point	A location which defines the distance between the reference points of the original and the image in the second column and second row, or in the case of only one column or one row, it is the distance between the reference points of the original and its neighboring image.	

ADD ARC

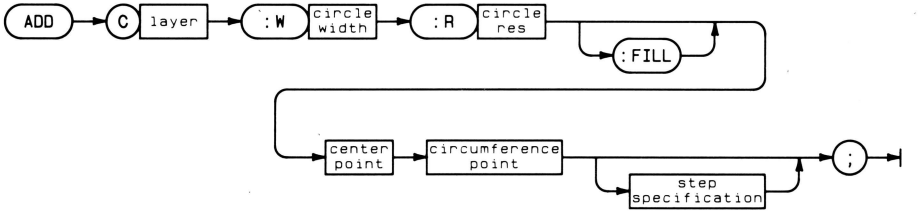
ADD ARC adds an arc to the drawing.



Item	Description/Default	Range Restrictions
A	The descriptor specifying an arc.	
layer	Specifies the arc layer.	1-255
:W arc width	Specifies the width of the arc outline.	Less than the radius
:R arc res	Specifies the angle in degrees between endpoints of the line segments which approximate the arc. This is the resolution. A 1° resolution produces the smallest arc segments. A 0° resolution will draw the arc as a smooth curve.	0-120°
:FILL	Specifies to fill an arc with width.	
first/second end point	End points of the arc.	
circumference point	A point on the circumference of the arc.	
step specification	Specifies to add an implicit step to the arc. See ADD for explanation.	

ADD CIRCLE

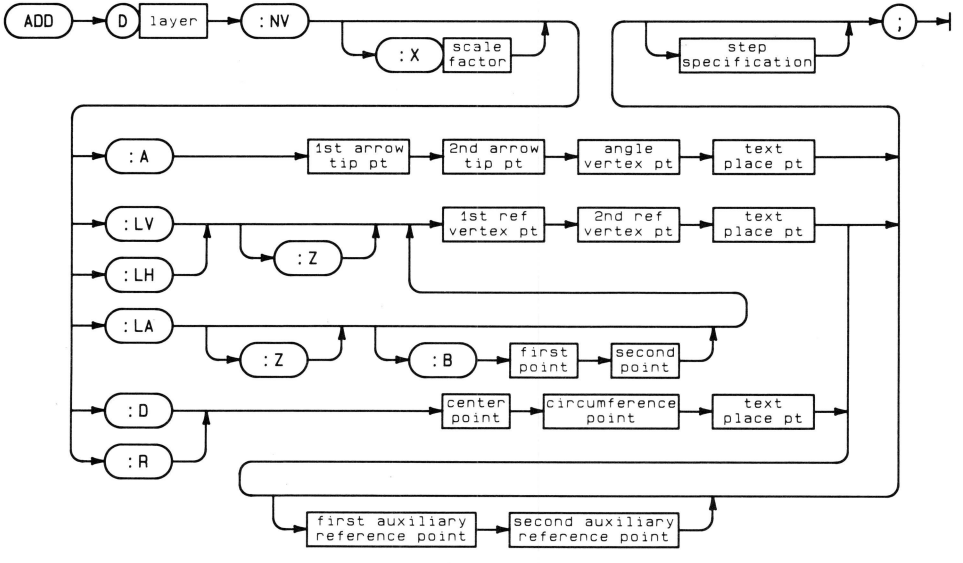
ADD CIRCLE adds a circle to the drawing.



Item	Description/Default	Range Restrictions
C	The descriptor specifying a circle.	
layer	Specifies the circle layer.	1-255
:W circle width	Specifies the width of the circle.	Less than the radius
:R circle res	Specifies the angle in degrees between endpoints of the line segments which approximate the circle. This is the resolution. A 1° resolution produces a circle consisting of 360 segments (360/1°). A 0° resolution draws the circle as a smooth curve.	0-120°
:FILL	Specifies to fill the circle.	
center point	Center point of the circle.	
circumference point	Point on circumference of the circle.	
step specification	Specifies to add an implicit step to the circle. See ADD for explanation.	

ADD DIMENSION

ADD DIMENSION adds dimensions to the drawing. There are four basic types of dimensions – line, angular, radius and diameter. Dimensions are made up of two parts: the dimension line(s) and the dimension value. The dimension value is associated text.



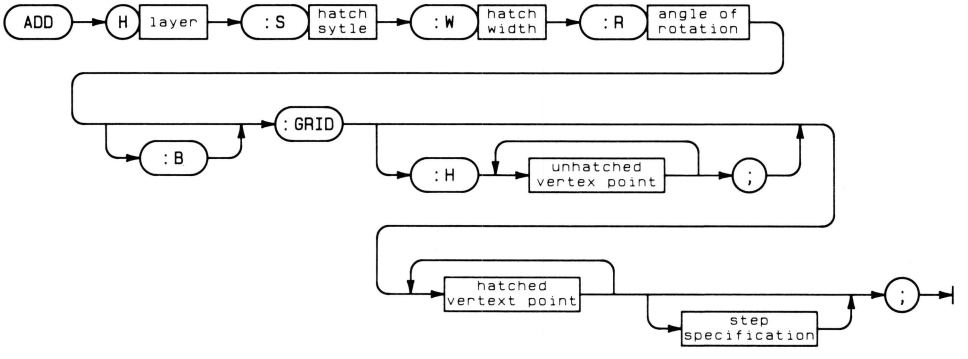
Item	Description/Default	Range Restrictions
D	The descriptor specifying a dimension.	
layer	Specifies the dimension layer.	1-255
:NV	Specifies that the dimension value exists and does not need to be evaluated.	
:X scale factor	Specifies the scale of the dimension arrowhead. This appears only if the scale factor is not 1.0.	
:A	Specifies an angular dimension.	
:LV	Specifies a vertical line dimension.	
:LH	Specifies a horizontal line dimension.	
:LA	Specifies an angular line dimension.	
:D	Specifies a diameter dimension	
:R	Specifies a radius dimension.	

(Table continued)

Item	Description/Default	Range Restrictions
:Z	Specifies that the linear dimension is a datum dimension.	
:B	Specifies a base line to which dimension arrow lines and values are parallel.	
1st/2nd arrow tip pt	For angular dimensions, specifies the location of the arrowhead tips.	
angle vertex pt	For angular dimensions, specifies the location of the vertex of the angle.	
text place point	The point in the center of the boundary rectangle enclosing the dimension value. However, for dimension or radius diameters with dimension values outside a circle, the text place point refers to the vertex of the leader line.	
1st/2nd ref vertex pt	For linear dimensions, specifies a point on an existing component.	
first/second point	Specifies two points on the base line.	
center point	For diameter or radius dimensions, specifies the center of a circle.	
circumference point	For diameter or radius dimensions, specifies an arbitrary point on the circumference of a circle. This is used for calculation of the radius or diameter of the circle.	
first/second auxiliary reference point	Specifies the geometric endpoints of the arc for diameter or radius dimensions. These points do not exist if the dimensioned component is a circle. While drawing linear dimensions in the isometric plane, the first and second auxiliary reference points specify the base line to which the extension lines are parallel.	
step specification	Specifies to add an implicit step. See ADD for explanation.	

ADD HATCH

ADD HATCH adds a hatched region to the drawing. The hatched region and unhatched areas within it are defined by points specified in the command.



Item	Description/Default	Range Restrictions
H	The descriptor specifying hatching.	
layer	Specifies the hatching layer.	1-255
:S hatch style	Style of hatching. 1 solid fill 2 single hatching 3 cross-hatching	1-3
:W hatch width	Width between hatching lines in user units.	
:R angle of rotation	Angle in degrees between hatching and the horizontal.	-90° to +90°
:B	Specifies to draw the boundary of the hatched region.	

(Table continued)

Item	Description/Default	Range Restrictions
:GRID	Sets the snapping mode to grid.	
:H	Specifies to leave unhatched areas.	
unhatched vertex point	Point on boundary of unhatched region (at least four points; last point equals the first).	
hatched vertex point	Point on boundary of hatched region (at least four points; last point equals the first).	
step specification	Specifies to add an implicit step. See ADD for explanation.	

Example

```
ADD H4 :S1 :W0.0000 :R0.0000 :GRID 50.0000,40.0000 80.0000,40.0000
  80.0000,110.0000 50.0000,110.0000 50.0000,40.0000;
```

This archive command represents solid hatching. The hatched region is bounded by the points (50,40), (80,40), (80,110), and (50,110). The first point, (50,40), is repeated to finish the hatched boundary.

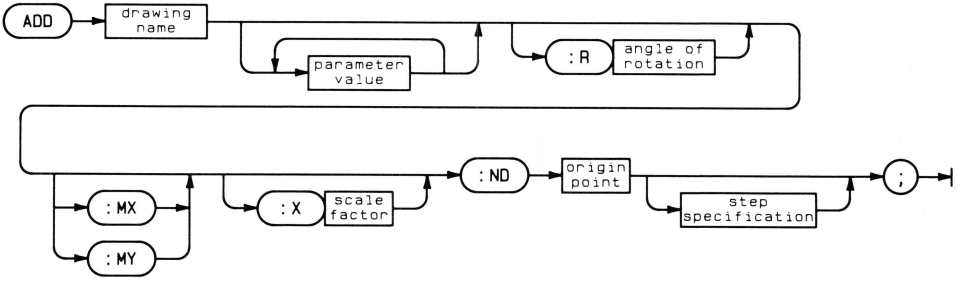
Example

```
ADD H5 :S2 :W5.0000 :R45.0000 :GRID
  :H 10.0000,10.0000 20.0000,10.0000 20.0000,20.0000 10.0000,20.0000
  10.0000,10.0000;
  0.0000,0.0000 100.0000,0.0000 100.0000,100.0000 0.0000,100.0000
  0.0000,0.0000;
```

This archive command represents hatching. The single hatching is added on layer 5, has a width of 5 inches, and is rotated 45°. The hatched region contains an unhatched area bounded by (10,10), (20,10), (20,20), and (10,20). The point (10,10) is repeated to complete the unhatched area. The hatched region is bounded by the points (0,0), (100,0), (100,100), and (0,100). The point (0,0) is repeated to complete the hatched region.

ADD instance

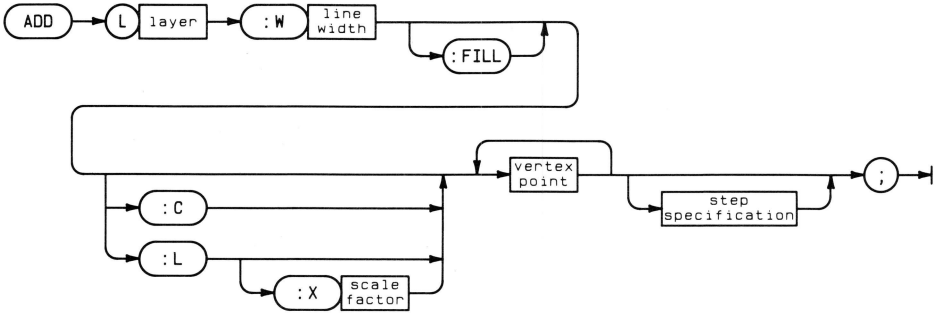
ADD instance adds an instance to the drawing.



Item	Description/Default	Range Restrictions
drawing name	The descriptor specifying an instance.	
parameter value	If a macro instance has any parameters, these parameters will appear.	
:R angle of rotation	Angle in degrees the instance is rotated about the instance origin. All components of an instance are displayed at the rotated angle. If a drawing was originally mirrored about both axes by :MXY, it will be archived by adding 180° to the original rotation angle.	-360° to +360°
:MX	Specifies that the instance added to the drawing will be mirrored about the X axis. An original :MY is archived as :MX with 180° added to the original rotation.	
:MY	Specifies that the instance added to the drawing will be mirrored about the Y axis. An original :MX with negative rotation is archived as :MY with 180° added to the original rotation.	
:X scale factor	Specifies the scale of the instance. For example, a scale factor of 0.5 causes the size of the instance to be half its original size.	
:ND	Specifies not to add default :AD text when adding the instance to the drawing.	
origin point	Point where the origin of the instance is to be located.	
step specification	Specifies to add an implicit step to the instance. See ADD for explanation.	

ADD LINE

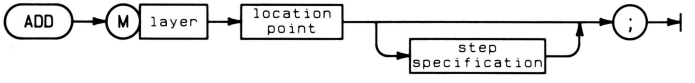
ADD LINE adds a line to the drawing.



Item	Description/Default	Range Restrictions
L	Specifies the line layer.	
layer	Layer to add the line.	1-255
:W line width	Specifies the width of the line. This width is displayed symetrically about the center of the line.	
:FILL	Specifies to fill the line.	
:C	Specifies a construction line.	
:L	Specifies an arrowhead is drawn at the first endpoint of a leader line.	
:X scale factor	Specifies the scale of a leader line arrowhead. This appears only if the scale factor is not 1.0.	
vertex point	Point on the line.	
step specification	Specifies to add an implicit step. See ADD for explanation.	

ADD MARKER

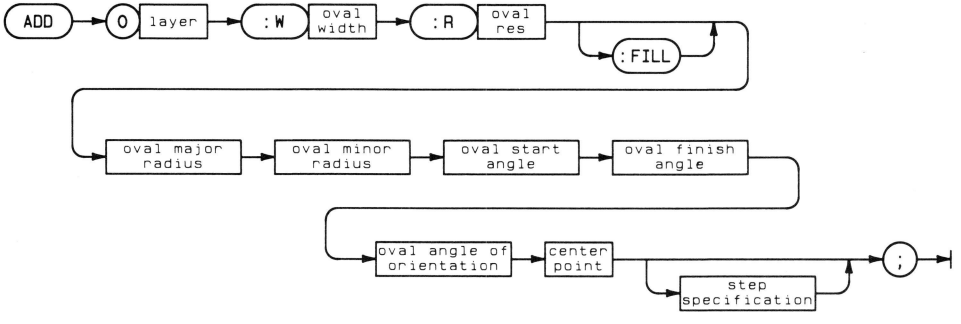
ADD MARKER adds a marker to the drawing.



Item	Description/Default	Range Restrictions
M	The descriptor specifying a marker component.	
layer	Specifies the marker layer.	1-255
location point	Location point of the marker.	
step specification	Specifies to add an implicit step. See ADD for explanation.	

ADD OVAL

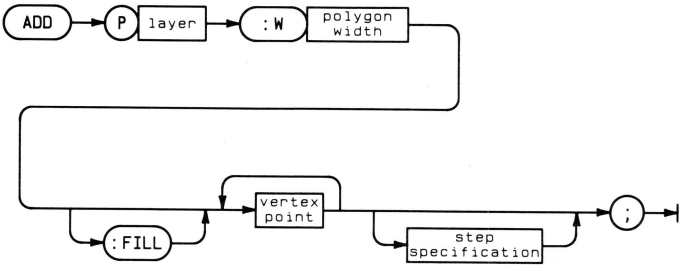
ADD OVAL adds an oval (ellipse) to the drawing.



Item	Description/Default	Range Restrictions
O	The descriptor specifying an oval (ellipse).	
layer	Specifies the oval layer.	1-255
:W oval width	Specifies the width of the oval outline.	
:R oval res	Specifies the angle in degrees between endpoints of the line segments which will approximate the oval. This is the resolution. A 1° resolution produces an oval with 360 line segments. A 0° resolution will draw the oval as a smooth curve.	0-120°
:FILL	Specifies to fill an oval. If the oval has width, the outline of the oval will be filled.	
oval major radius	The oval's major radius length.	
oval minor radius	The oval's minor radius length.	
oval start angle	The angle measured counterclockwise from the major axis of the oval to the starting angle of the oval. This angle is 0° for complete ellipses.	
oval finish angle	The angle measured counterclockwise from the major axis of the oval to the finish point of the oval. This angle is 360° for complete ellipses.	
oval angle of orientation	The angle of rotation measured counterclockwise from an imaginary horizontal line to the oval's major axis.	
center point	Center point of the oval.	
step specification	Specifies to add an implicit step. See ADD for explanation.	

ADD POLYGON

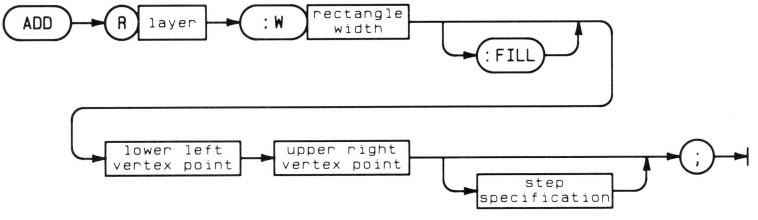
ADD POLYGON adds a polygon to the drawing.



Item	Description/Default	Range Restrictions
P	The descriptor specifying a polygon.	
layer	Specifies the polygon layer.	1-255
:W polygon width	Specifies the polygon outline will have width.	
:FILL	Specifies to fill the polygon. If the polygon has width, the outline of the polygon will be filled.	
vertex point	Point on boundary of the polygon (at least four points; last point equals the first).	
step specification	Specifies to add an implicit step to the polygon. See ADD for explanation.	

ADD RECTANGLE

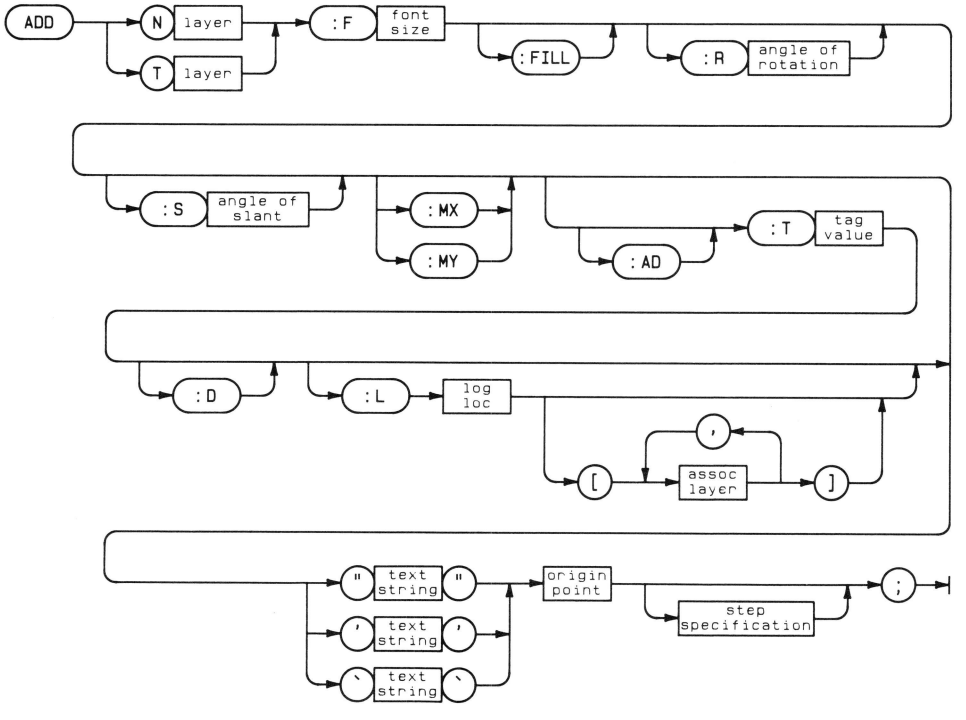
ADD RECTANGLE adds a rectangle to the drawing.



Item	Description/Default	Range Restrictions
R	The descriptor specifying a rectangle.	
layer	Specifies the rectangle layer.	1-255
:W rectangle width	Specifies that the outline of the rectangle will have width.	
:FILL	Specifies to fill the rectangle. If the rectangle has width, the outline of the rectangle will be filled.	
lower left/upper right vertex point	Location of opposite corners of the rectangle.	
step specification	Specifies to add implict step to the rectangle. See ADD for explanation.	

ADD TEXT or NOTE

ADD TEXT adds characters with width that can be filled. ADD NOTE adds line-thickness characters to the drawing.



Item	Description/Default	Range Restrictions
N	The descriptor specifying line thickness characters.	
T	The descriptor specifying characters with width.	Only uppercase letters (A-Z)
layer	Specifies the text or note layer.	1-255
:F font size	Height of the characters, in user units.	
:FILL	Specifies to fill text characters.	

(Table continued)

Item	Description/Default	Range Restrictions
:R angle of rotation	Angle in degrees the characters are rotated. The angle is measured counterclockwise from an imaginary horizontal line. An original :MXY is archived by adding a 180° to the original rotation angle.	-360° to +360°
:S angle of slant	Angle in degrees of the text slant. The angle is measured from the vertical (clockwise is positive).	-75° to +75°
:MX	Specifies to mirror the characters about the X axis. An original :MY is archived as :MX with 180° added to the original rotation.	
:MY	Specifies to mirror the characters about the Y axis. An original :MX with negative rotation is archived as :MY with 180° added to the original rotation.	
:AD	For associated text, specifies to associate the text with the drawing being edited.	
:T	Specifies to include a tag value with the asociated text. This is for use in post-processing applications to specify a type or class of associated text.	
tag value	The tag value of the associated text.	-32768 to 32767
:D	For associated text, specifies to display the added text.	
:L	For associated text, specifies to include a logical location for the associated text.	
assoc layer	For associated text, specifies which layers the logical location is connected to.	
log loc	For associated text, selects the logical location of the associated text.	
text string	Text to be added. Delimiters can be double quotes (''), single quotes ('), or reverse single quotes (‘).	
origin point	Origin of the characters to be added. This is the lower left corner of boundary rectangle.	
step specification	Specifies to add an implicit step. See ADD for explanation.	

BEGIN_ASSOC

BEGIN_ASSOC is the first of a series of commands which associate text to a component. The command series concludes with END_ASSOC. Only ADD TEXT, ADD NOTE or other BEGIN_ASSOC - END_ASSOC series can be nested within the BEGIN_ASSOC and END_ASSOC pair. Text or Notes can be associated to other text or notes by nesting one BEGIN_ASSOC - END_ASSOC pair within another.

Example

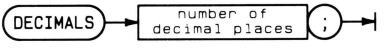
```
ADD M3 20.0000,60.0000;
  BEGIN_ASSOC
  ADD N3 :F10.0000 :T1001 "NOTE" 30.0000,70.0000;
    BEGIN_ASSOC
    ADD N3 :F10.0000 :T1002 "FIRST ASSOCIATED TO NOTE" 50.0000,90.0000;
    ADD N3 :F10.0000 :T1002 "SECOND ASSOCIATED TO NOTE" 50.0000,110.0000;
    END_ASSOC
  ADD N3 :F10.0000 :T1001 "NOTE2" 30.0000,40.0000;
  END_ASSOC
```

This series of archive commands represents several associated notes. `NOTE` is placed at (30,70) and associated to the marker. `NOTE` has two notes associated to it, `FIRST ASSOCIATED TO NOTE` and `SECOND ASSOCIATED TO NOTE`. The first note is located at (50,90) and the second is located at (50,110).

In addition, a second note, `NOTE2`, is also added and associated to the marker. This note is located at (30,40).

DECIMALS

DECIMALS specifies the number of decimal places for all numbers except dimensions. DECIMALS is listed in the archive file if :P was specified during archiving.



Item	Description/Default	Range Restrictions
number of decimal places	Number of decimal places numerical output is to have.	0-10

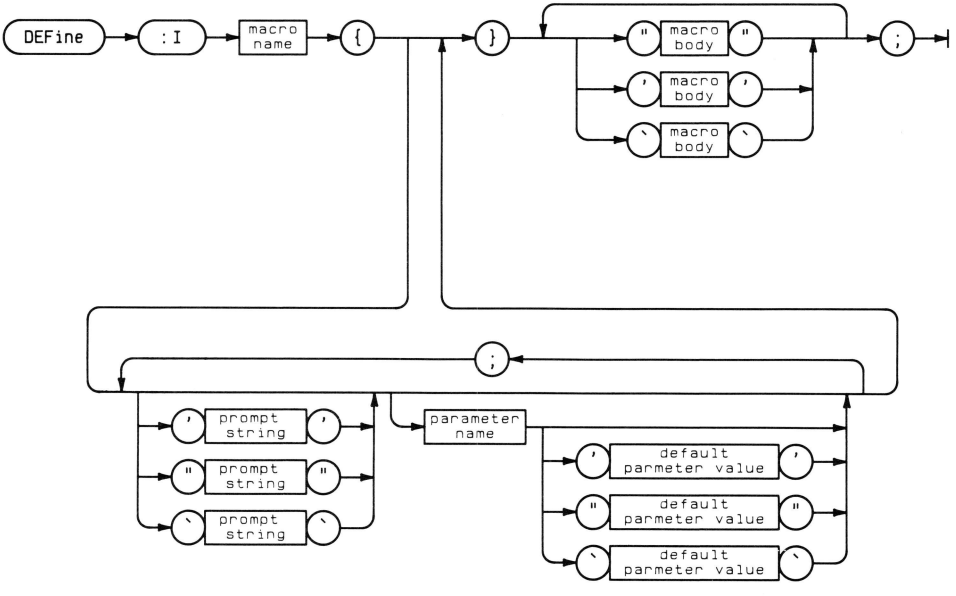
Example

```
DECIMALS 3;
```

This archive command reflects the DECIMALS default when the drawing was archived. It specifies that numbers are correct to three decimal places.

DEFINE

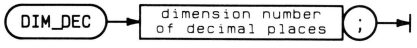
DEFINE specifies a macro instance in the archive file. A macro instance textually describes an instance by using Graphics Editor commands and/or other macros. For a complete explanation of macro instances, see DEFINE in the Syntax Reference and the example in the Archive Reference introduction.



Item	Description/Default	Range Restrictions
:I	Specifies that a macro instance is to follow in the archive file. This is the only use of DEFINE in an archive file.	
macro name	Name of the macro instance.	20 characters maximum
prompt string	A user-definable prompt for any parameter in the macro.	80 characters maximum, 12 parameters maximum
parameter name	A placeholder for data to be entered with the newly created command.	20 characters maximum, 12 parameters maximum
parameter def value	Default value for the parameter if no data is entered with the newly-created command.	20 characters, 12 parameters maximum
macro line	A string of commands and parameters which form the macro or a line of the macro.	Individual strings are limited to 250 characters maximum, but you can have an unlimited number of these individual strings.

DIM_DEC

DIM_DEC specifies the number of decimal places of dimension values. DIM_DEC is listed in the archive file if :P was specified during archiving. Although similar to DIM_DEC, DECIMALS specifies the number of decimal places for all numbers except dimensions.



Item	Description/Default	Range Restrictions
dimension number of decimal places	Number of decimal places of dimension values.	0-10

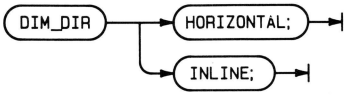
Example

```
DIM_DEC 3;
```

This archive command reflects the DIM_DEC default when the drawing was archived. It specifies that all dimensions are output to three decimal places.

DIM_DIR

DIM_DIR specifies the direction and orientation of dimension values and tolerance values. DIM_DIR is listed in the archive file if :P was specified during archiving.



Item	Description/Default	Range Restrictions
HORIZONTAL	Specifies to display dimensions values horizontally, regardless of the orientation of the dimension line.	
INLINE	Specifies to display dimension values parallel to the dimension arrows. If the dimension arrow is vertical, the dimension value will be horizontal.	

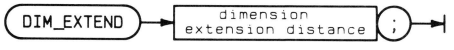
Example

```
DIM_DIR HORIZONTAL;
```

This archive command reflects the DIM_DIR default when the drawing was archived. It specifies that all dimension values are displayed horizontally.

DIM_EXTEND

DIM_EXTEND specifies the distance between dimension extension lines and the object being dimensioned. The extension lines also extend past the dimension line by the same distance. DIM_EXTEND is listed in the archive file if :P was specified during archiving.



Item	Description/Default	Range Restrictions
dimension extension distance	Default distance for the dimension extension.	0 to 10^9 divided by user units per system grid point

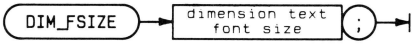
Example

```
DIM_EXTEND 10;
```

This archive command reflects the DIM_EXTEND default when the drawing was archived. It specifies that all dimensions lines are extended 10 user units.

DIM_FSIZE

DIM_FSIZE specifies the font size of dimension labels. DIM_FSIZE is listed in the archive file if :P was specified during archiving.



Item	Description/Default	Range Restrictions
dimension text font size	Default font size for dimension text.	>0

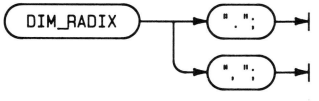
Example

```
DIM_FSIZE  10;
```

This archive command reflects the DIM_FSIZE default when the drawing was archived. It specifies that the dimension font size is 10 user units.

DIM_RADIX

DIM_RADIX specifies the radix (delimiter) of dimension values. The radix can be a period or comma. (European convention separates integers from decimals with a comma rather than a period; for example, 10,000 instead of the US convention of 10.000.) DIM_RADIX is listed in the archive file is :P was specified during archiving.



Example

```
DIM_RADIX ",";
```

This archive command reflects the DIM_RADIX default when the drawing was archived. It specifies that the radix is a comma.

DIM_SCALE

DIM_SCALE specifies the scale value for dimensions. DIM_SCALE is listed in the archive file if :P was specified during archiving.

Item	Description/Default	Range Restrictions
dimension scale value	Specifies to multiply the entered dimension value by the given default value.	

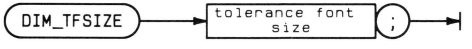
Example

```
DIM_SCALE ,1;
```

This archive command reflects the DIM_SCALE default when the drawing was archived. It specifies that dimension values are one-tenth scale.

DIM_TFSIZE

DIM_TFSIZE specifies the font size of dimension tolerance values. DIM_TFSIZE is listed in the archive file if :P was specified during archiving.



Item	Description/Default	Range Restrictions
tolerance font size	Species the font size of dimension tolerance values.	

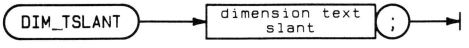
Example

```
DIM_TFSIZE 0.0938;
```

This archive command reflects the DIM_TFSIZE default when the drawing was archived. It specifies that the tolerance font size is 0.0938 user units.

DIM_TSLANT

DIM_TSLANT specifies the slant of dimension values. DIM_TSLANT is listed in the archive file if :P was specified during archiving.



Item	Description/Default	Range Restrictions
dimension text slant	Specifies the slant of dimension values	-75° to +75°

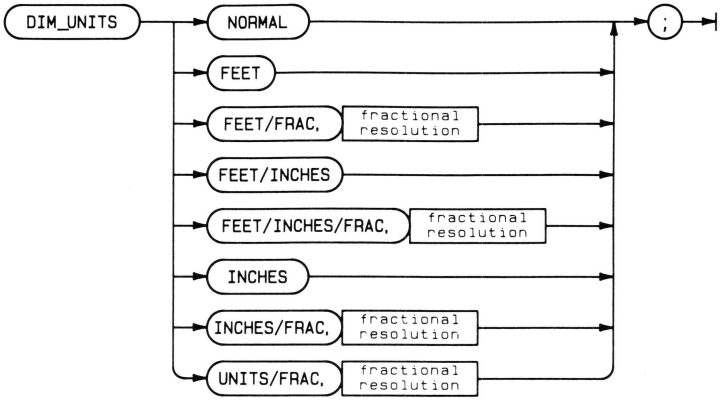
Example

```
DIM_TSLANT 30;
```

This archive command reflects the DIM_TSLANT default when the drawing was archived. It specifies that the slant of dimension values is 30° to the right.

DIM_UNITS

DIM_UNITS specifies the units of line, radius, and diameter dimensions. DIM_UNITS is listed in the archive file if :P was specified during archiving.



Item	Description/Default	Range Restrictions
NORMAL	Specifies to only display dimension values in decimal units.	
FEET	Specifies to only display dimension values in FEET.	
FEET/FRAC	Specifies to only display dimension values in FEET and FRACTIONS of feet.	
FEET/INCHES	Specifies to only display dimension values in FEET and INCHES.	
FEET/INCHES /FRAC	Specifies to only display dimension values in FEET, INCHES, and FRACTIONS of an inch.	
INCHES	Specifies to only display dimension values in INCHES.	
INCHES/FRAC	Specifies to only display dimension values in INCHES and FRACTIONS of an INCH.	
UNITS/FRAC	Specifies to only display dimension values in the current units defined and fractions of those units.	

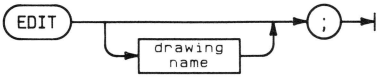
Example

```
DIM_UNITS FEET/INCHES/FRAC,64;
```

This archive command reflects the DIM_UNITS default when the drawing was archived. It specifies that the dimension units are feet, inches, and fractions (1/64) of an inch.

EDIT

EDIT has two purposes in an archive file: to clear the screen for retrieving an archived drawing and to begin a cycle of commands for re-creating an archived instance or drawing.



Item	Description/Default	Range Restrictions
drawing name	Specifies the name of the drawing which will appear on the screen.	maximum 8 characters

Example

```
EDIT;
```

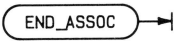
This archive command clears the drawing screen.

```
EDIT part1;
```

This archive command begins the series of commands that re-create `part1`.

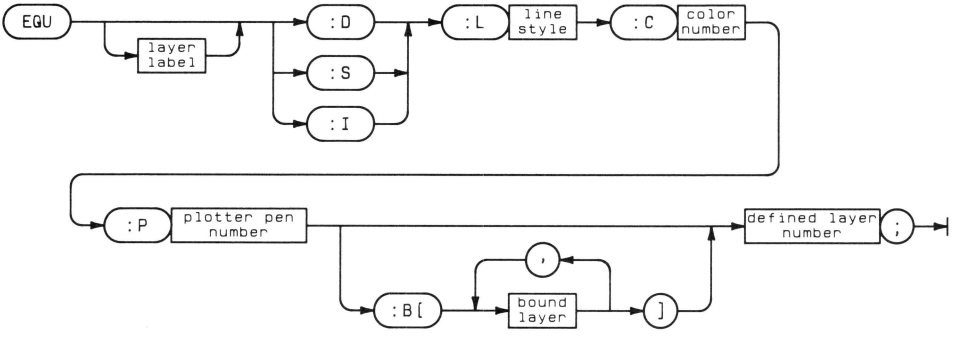
END_ASSOC

END_ASSOC is the ending delimiter that specifies text will be associated to a component. See BEGIN_ASSOC for explanation.



EQUATE

EQUATE defines each layer of the drawing area in the Graphics Editor. Layer 0 is reserved by the system for displaying instances of library parts and is always defined by the system. EQUATE is listed in the archive file if :P was specified during archiving.



Item	Description/Default			Range Restrictions
layer label	A note which describes the contents of the layer.			
:D	Specifies that the layer is a detail layer.			
:S	Specifies that the layer is a symbolic layer.			
:I	Specifies that the layer is an interconnect layer.			
:L line style	Specifies the line style by number.			1-8
	Line Type	**Number**	**Result**	
	solid	1	———	
	dashed	2	— — — —	
	long dash	3	— —— ——	
	dot center	4	——— . ———	
	dash center	5	——— — ———	
	phantom	6	—— — — ——	
	long dotted	7	. . . . .	
	dotted	8	.....	

(Table continued)

Item	Description/Default	Range Restrictions
:C color number	Specifies the color to display the components on the layer being defined. The color is specified by a number: White 1 Red 2 Yellow 3 Green 4 Cyan 5 Blue 6 Magenta 7 Black 8 Olive Green 9 Aqua 10 Royal Blue 11 Maroon 12 Brick Red 13 Orange 14 Brown 15	
:P plotter pen number	Plotter pen number with which to plot components in this layer.	0-8
:B	Specifies which other layers are bound to the layer.	
bound layer	Specifies the layers which are bound to the particular layer.	0-255
defined layer number	Number of the layer being defined.	0-255

Example

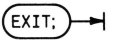
```
$$                Layer   Line
$$  Label         Type    Type    Color   Pen    Layer
EQU INSTBOUN      :D      :L8     :C1     :P1     0;
EQU NPUT          :D      :L1     :C3     :P3     1;
```

This series of archive file commands reflect two of the layer definitions when the drawing was archived. Layer 0 is reserved by the system for displaying instances of library parts. Layer 1 is defined to be a detail layer. Components entered on this layer are drawn in yellow (on systems with a color CRT) and with a solid line type. Components on layer 1 are plotted using pen number 3. The label NPUT is assigned to this layer.

For an example of EQUATE used in a process file, see the Archive Reference introduction.

EXIT

EXIT indicates the end of a process file when it follows a series of EQUATE commands. EXIT also indicates the end of an unnamed instance addition when it follows a series of ADD commands.



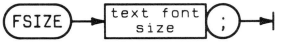
$FILES

$FILES is followed by a list of drawings and instances which will be created when the archive file is retrieved. During retrieval, $FILES removes any drawings and instances in the destination volume or memory which have the same name as the drawings and instances in the list. For an example of $FILES, see the Archive Reference introduction.

Item	Description/Default	Range Restrictions
drawing name	Name of drawings or instances to be removed from the destination volume and/or memory.	Maximum 8 characters

FSIZE

FSIZE specifies the font size for all notes and text. DIM_FSIZE specifies the font size of dimension values. FSIZE is listed in the archive file if :P was specified during archiving.



Item	Description/Default	Range Restrictions
text font size	Default font size for notes and text.	>0

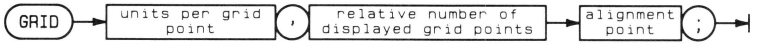
Example

```
FSIZE 5;
```

This archive command reflects the FSIZE default when the drawing was archived. It specifies that the default font size of text and notes is 5 user units.

GRID

GRID sets the user grid to maximum resolution before adding components during the retrieval process and after the components have been retrieved, resets the grid to the resolution of the original drawing. In addition, GRID is included in the process file if :P was specified during archiving.



Item	Description/Default	Range Restrictions
units per grid point	Number of user units between consecutive user grid points. User units are specified in the process file with the UNITS command.	
relative number of displayed grid points	Determines the number of user grid points that are displayed. If, for example, the number 5 is entered, every fifth user grid point is displayed.	
alignment point	Shifts the grid such that a grid point is located exactly on the alignment point. The default alignment point is the drawing origin 0,0.	

Example

```
GRID 0,0001,1 0,0;
```

This archive command specifies a grid with maximum resolution for units of ten-thousanths of an inch. The grid has a point located at (0,0).

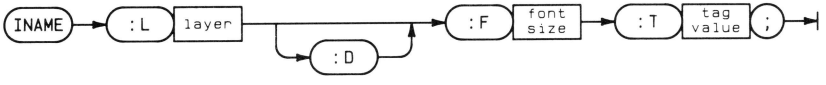
Example

```
GRID  0,1250,2 0,0000,0,0000;
```

This archive command represents the grid setting when the drawing was archived. It specifies a grid point every 0.1250 inch and that every second grid point is displayed. A grid point is located at (0,0).

INAME or TNAME

INAME defines the tag value, font size, layer, and display option defaults when transferring INAME associated text from EGS-45 to HP EGS. This is necessary because HP EGS requires these default values for associated text. INAME is listed in the archive file if :P was specified during archiving. TNAME is the same as INAME.



Item	Description/Default	Range Restrictions
:L layer	Specifies the default layer for INAME/TNAMEs entered from EGS-45.	1-255
:D	Specifies that INAME text will be displayed.	
:F font size	Specifies the default font size for INAME/TNAMEs entered from EGS-45 archive files.	
:T tag value	Specifies the default tag value for INAME/TNAMEs entered from EGS-45 archive files.	

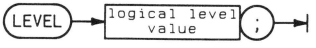
Example

```
INAME :L7 :D :F0.3000 :T1006;
```

This example specifies the default parameters required when transferring EGS-45 associated text to HP EGS. These parameters are layer 7, a font size of 0.3 inch, and a tag value of 1006. The text is displayed. TNAME could have been substituted for INAME.

LEVEL

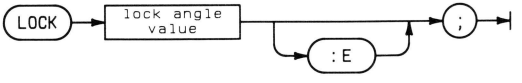
LEVEL assigns the logical level of a drawing. LEVEL is listed in the archive file if :P was specified during archiving.



Item	Description/Default	Range Restrictions
logical level value	Specifies the level depth of the drawing. The default is 1.	-32768 to 32767

LOCK

LOCK sets the lock angle to 0° to disable lock angle enforcement before adding components during the retrieval process and after the components have been retrieved, resets the lock angle to the value last used in the drawing. In addition, LOCK is listed with the process file if :P was specified during archiving.



Item	Description/Default	Range Restrictions
lock angle value	The lock angle in degrees.	0-180° in .0001° increments
:E	Specifies lock angle enforcement. :E only appears with LOCK in the process file and is disabled by LOCK 0.0; before any components are added.	

Example

```
LOCK 0.0;
```

This archive command sets the lock angle value to 0° for maximum resolution during component retrieval.

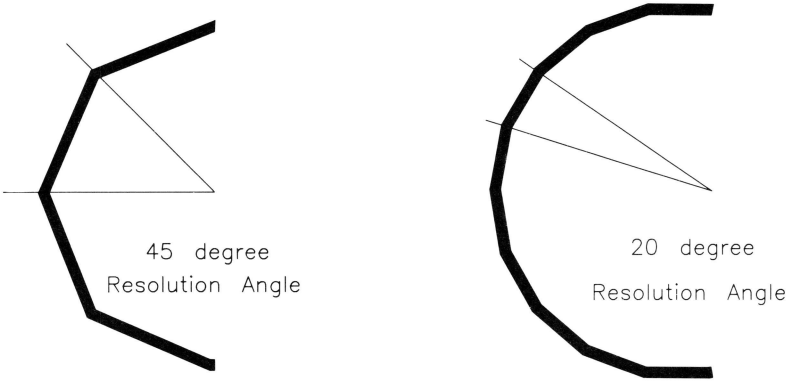
RESOLUTION

RESOLUTION sets the curve resolution of arcs, circles, and ovals. RESOLUTION is listed in the archive file if :P was specified during archiving.

Item	Description/Default	Range Restrictions
resolution value	Specifies the resolution for displaying and plotting arcs, circles, and ovals. If none is specified with the ADD command for a particular component, the system will use this resolution.	0-120°

The Resolution Angle

The following illustrations show two arcs with different resolution angles.



The smaller the resolution angle, the smaller the line segments which approximate the circle or arc. For a circle or oval, the number of line segments equals 360 divided by the resolution angle. Therefore, a 1° resolution angle would produce a circle or oval that is approximated by 360 segments (360/1°).

A 0° resolution will draw a smooth curve.

SAVE

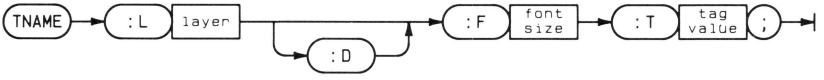
SAVE saves the drawing or named graphical instance previously listed in the archive file after EDIT. The drawing is saved on the destination volume. For an example of the use of SAVE, see the Archive Reference introduction.

SHOW

SHOW #E specifies that all of the components on all of the layers are displayed and modifiable.

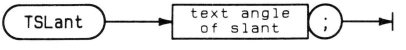
TNAME

See INAME for explanation.



TSLANT

TSLANT specifies the slant of all notes and text. The slant of dimension values is specified by DIM_TSLANT. TSLANT is listed in the archive file if :P was specified during archiving.



Item	Description/Default	Range Restrictions
text slant	Specifies the slant of text and notes.	-75° to +75°

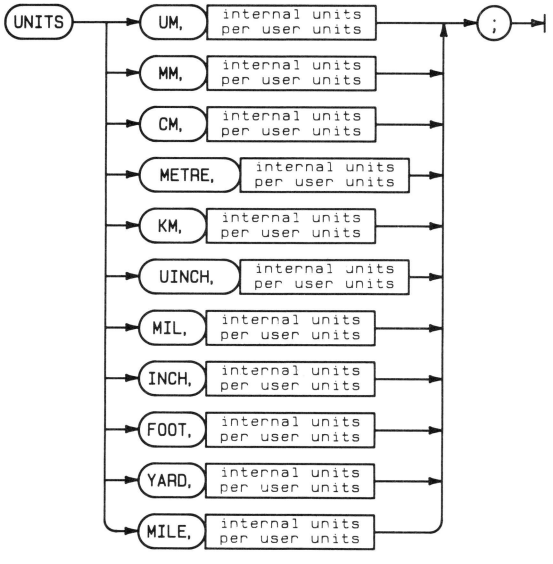
Example

```
TSLANT 30.0000;
```

This archive command represents the text and note slant when the drawing was archived. It specifies that notes and text have a slant of 30° to the right.

UNITS

UNITS assigns the physical distance between system grid points, thereby determining the drawing resolution. UNITS is listed in the archive file if :P was specified during archiving.



Item	Description/Default	Range Restrictions
UM	Specifies micrometres as the units.	
MM	Specifies millimetres as the units.	
CM	Specifies centimetres as the units.	
METRE	Specifies metres as the units.	
KM	Specifies kilometres as the units.	
UINCH	Specifies micro-inches as the units.	
MIL	Specifies mils as the units.	
INCH	Specifies inches as the units.	
FOOT	Specifies feet as the units.	
YARD	Specifies yards as the units.	
MILE	Specifies miles as the units.	
internal units per user units	Specifies the number of system grid points that are equal to one user units.	

Example

```
UNITS INCH,1000;
```

This archive command reflects the current units when the drawing was archived. It specifies that 1000 system grid points is equal to 1 inch. Therefore, each grid point is 0.001 inch apart. Since an xy location can only be entered on a grid point, the resolution is limited to 0.001 inches.

WINDOW

WINDOW specifies the coordinates necessary to fit the drawing in the screen area.

Item	Description/Default	Range Restrictions
lower left point, upper right point	The screen coordinates that define the rectangular area enclosing the drawing.	

Example

```
WINDOW 0.0000,0.0000 100.0000,100.0000;
```

This archive command fits the drawing in the rectangular area defined by the points (0,0), (0,100), (100,0), and (100,100).

Subject Index

a

Aborting commands while in the text editor . . . . . **M** 31
Access . . . . . **M** 127
Activating tablet menus . . . . . **S** 201, **L** 291, **U** 93
Active:
  Commands . . . . . **S** 5
  Drawing area . . . . . **U** 89
ACTUAL:
  DRILL (Photoplot/Drill) . . . . . **U** 316
  PHOTOPLOT (Photoplot/Drill) . . . . . **U** 316
AdBox . . . . . **L** 55
AdCom . . . . . **L** 55
ADD . . . . . **S** 7
ADD ARC . . . . . **S** 10
ADD ARC (archive) . . . . . **S** 250
ADD (archive) . . . . . **S** 248–249
ADD CIRCLE . . . . . **S** 14
ADD CIRCLE (archive) . . . . . **S** 251
ADD DIMENSION . . . . . **S** 18
ADD DIMENSION (archive) . . . . . **S** 252–253
ADD HATCH . . . . . **S** 24–28
ADD HATCH (archive) . . . . . **S** 254–255
ADD instance . . . . . **S** 30–31
ADD instance (archive) . . . . . **S** 256–257
ADD LINE . . . . . **S** 32–37
ADD LINE (archive) . . . . . **S** 258
ADD MARKER . . . . . **S** 39
ADD MARKER (archive) . . . . . **S** 259
ADD NOTE . . . . . **S** 48–52

I *Installing HP EGS 2.2*
S *HP EGS 2.2 Syntax Reference*
L *Learning HP 2.2*
U *Understanding HP EGS 2.2*
M *Managing and Editing Files with HP EGS 2.2*

ADD NOTE (archive) . . . . . S 264–265
ADD OVAL . . . . . S 40–42
ADD OVAL (archive) . . . . . S 260–261
ADD POLYGON . . . . . S 44–45
ADD POLYGON (archive) . . . . . S 262
ADD RECTANGLE . . . . . S 46–47
ADD RECTANGLE (archive) . . . . . S 263
ADD TEXT . . . . . S 48–52
ADD TEXT (archive) . . . . . S 264–265
ADD* . . . . . L 23
Adding:
Angular dimension . . . . . L 122
Arcs . . . . . L 40, L 107, S 10
Bevel . . . . . S 34
Beveled face . . . . . L 86
Board blank outline . . . . . L 252
Bolt . . . . . L 134
Bolt head . . . . . L 144
Capacitors and resistors . . . . . L 183, L 239
Center lines . . . . . L 108, L 109, L 118
Circle . . . . . L 30, S 14
Connections . . . . . L 194
Construction circles . . . . . L 106
Construction lines . . . . . L 59, L 105, L 112, S 33
Diameter holes . . . . . L 108
Diametral dimensions . . . . . L 123
Dimension notes . . . . . L 124
Edge connector . . . . . L 251
Flange . . . . . L 120
Flange part . . . . . L 132, L 148
Front view dimensions . . . . . L 125
Hole . . . . . L 88
Isometric rectangle . . . . . L 84
Keepout area . . . . . L 243
Leader lines . . . . . S 33, L 151
Linear dimensions . . . . . L 168
Logical ports . . . . . L 189
Material list . . . . . L 163
Multivertex lines . . . . . L 43
Note . . . . . S 48–52, L 94
Nut and washer items . . . . . L 149

Outline . . . . . L 197
Pad set . . . . . L 240
Part type . . . . . L 188, L 245
Physical ports . . . . . L 190, L 246
Polygon . . . . . L 32, S 44–45
Primitive . . . . . L 23
Radial dimensions . . . . . L 123
Reference designator . . . . . L 188, L 245
Roof line . . . . . L 28
Semicircle and tangent lines . . . . . L 106
Solder resist . . . . . L 229
Source voltage input . . . . . L 196
Surface finish symbol . . . . . L 128
Text while in the text editor . . . . . M 29, M 102
Tooling marks . . . . . L 264
Traces . . . . . L 263
Vertex . . . . . L 60
Vertical dimensions . . . . . L 126
Washer and nut . . . . . L 136
Adjust . . . . . M 93
Adjusting margins . . . . . M 45, M 93
Aligning a part . . . . . L 187, L 245
Alignment of grid with GRID command . . . . . S 140
*ALL . . . . . L 44, L 102
Alpha:
Dump . . . . . S 107
Screen . . . . . L 16, U 7
Analog parts . . . . . L 286
*ANG . . . . . L 28, L 93
*ANGL . . . . . L 41
Angle, lock angle warning . . . . . S 164
*ANGS . . . . . L 40
Angular dimension . . . . . L 122, S 19
Annotating:
Block . . . . . L 90
Flange drawing . . . . . L 120
Annotatn mode . . . . . L 101
Apertures . . . . . U 289

Application notes . . . . . U Appendix
Applications, user-written . . . . . U 46
ARC, ADD . . . . . S 10
Arcs, adding . . . . . L 40, L 107
Arc methods . . . . . L 40
ARCHIVE . . . . . L 77, S 53–55, U 352
Archive file:
  Commands used in . . . . . S 248–300
  Description . . . . . S 240–246, U 24–27
  $FILES . . . . . S 243
  Process file in . . . . . S 240
  Retrieving drawings and instances with EDIT . . . . . S 243
  Retrieving macro instances with DEF :I . . . . . S 244
  Using to transport data . . . . . U Appendix
Archive Reference . . . . . S 239–300
Archived drawings, retrieving with RETRIEVE . . . . . S 198
Archiving drawings . . . . . U 145
AREA . . . . . S 56–57
AREA, isometric . . . . . S 57
AREA? . . . . . L 53
Array:
  Creating with STEP . . . . . S 208–211
  Step specification . . . . . S 8
ASCII ordinals . . . . . U 103
Assembler . . . . . M 73
Assembling parts . . . . . L 132
Assembly drawing:
  Annotating . . . . . L 147
  Creating . . . . . L 275
  Material listing . . . . . L 156
Associated text:
  :AC text . . . . . S 50, U 20–21, U 144
  :AD text . . . . . S 50, U 20–21, U 144
  Description . . . . . U 20–21
  Drawing . . . . . U 80, U 90
  Tag values . . . . . U 144, U 306
Attribute data . . . . . U 279, U 312–315
ATTRIBUTES (Photoplot/Drill) . . . . . U 312
AUTOFLASH (Photoplot/Drill) . . . . . U 295, U 315

b

:B, used to defined a hatched region . . . . . S 25
BACK_ANNOTATE . . . . . S 58–61, U 182–183
Back annotation file . . . . . U 195, U 182–219
Back-annotating a schematic drawing . . . . . U 206
BACK SPACE . . . . . L 5, L 6
BACK SPACE . . . . . L 5, L 6
Backing up system on CS/80 tapes . . . . . M 16
Backup (^) . . . . . S 62
Bad sector . . . . . M 130
*BASE . . . . . L 168
Base line, dimension . . . . . S 19–20
Basic layers . . . . . L 229
BEGIN_ASSOC . . . . . S 266
Bevel, ADD LINE . . . . . S 34
Beveled face, adding . . . . . L 86
Bill of materials, creating . . . . . L 214
BK, backup for HP EGS modules . . . . . L 26
*BLNK . . . . . L 103, L 147
Block letters (text) . . . . . S 48–52
Board blank outline, adding . . . . . L 252
Bolt head, adding . . . . . L 144
Bolt, adding . . . . . L 134
Boot-up display . . . . . I 26
BOTH . . . . . L 44
Breaking up large files . . . . . M 18
BUBBLE . . . . . L 103
Building:
  Control file . . . . . U 42
  Macro file . . . . . U 62
  Process file . . . . . U 52
  Screen menu . . . . . U 57
  Start files . . . . . U 48
  Tablet menu . . . . . U 88
  Volume search tables . . . . . U 50
Bus addresses, setting on a peripheral . . . . . I 18–20
Bus notation (Connection Lister) . . . . . U 141
Busline . . . . . L 179
Buslines for connections . . . . . L 218
Bytes, seeing the number of at boot-up . . . . . I 26

C

*C&C . . . . . L 30
*C&Rs . . . . . L 102
Calculating expressions with EVALUATE . . . . . S 120
CALIBRATE . . . . . S 63–65
Capacitors and resistors, adding . . . . . L 183, L 239
CATALOG . . . . . S 66–67
Center lines and fillets, deleting . . . . . L 139
Center lines, adding . . . . . L 108, L 109, L 118
CENTR . . . . . L 38
Certificate, right-to-use (codeword) . . . . . I 69, U 283, U 346
Chain copying . . . . . S 70–75
*CHAM . . . . . L 28
Chamfer:
  ADD LINE . . . . . S 34
  Dimensioning . . . . . L 127
CHANGE . . . . . S 68–69, L 80
Change . . . . . M 60, M 131
Changing:
  Color . . . . . L 51
  Default volume . . . . . M 54, M 150
  File names . . . . . M 60, M 131
  File types . . . . . M 61, M 157
  Parts file . . . . . L 157
  Tolerancing option . . . . . L 93
  Width . . . . . L 51
Character cell . . . . . U 102
Character code, converting . . . . . M 20
CIRCLE:
  ADD . . . . . S 14
  Methods . . . . . L 30
CLR -> END . . . . . L 5
CLR I/O . . . . . L 5
CLR LN . . . . . L 5, L 6
*CMPT . . . . . L 103
*CNTR . . . . . L 102

Codeword:
Installation utility . . . . . I 74–77
Installing one codeword . . . . . I 76–77
Installing several codewords at once . . . . . I 77–79
Obtaining . . . . . I 74–76
Right-to-use certificate . . . . . I 73, U 283, U 346
Collecting part information . . . . . L 147
Collections, building, modifying, or destroying with GATHER . . . . . S 127–132
COLOR (SHOW* command) . . . . . L 44
Color:
Changing . . . . . L 51
Cursor . . . . . S 77–79
Defining component colors with EQUATE . . . . . S 119
Grid . . . . . S 139
Of each entry . . . . . U 58
Screen menu . . . . . S 168
Command entry:
Editing keys . . . . . U 9–10
Files . . . . . U 3
Graphics input device . . . . . U 3
Keyboard . . . . . U 3
Screen menu . . . . . U 3
Tablet menu . . . . . U 3
Command types:
Commands to define other commands and macros . . . . . U 32–33
Current . . . . . S 5, U 11–12
Default . . . . . U 32–33
Drawing creation and editing . . . . . U 32–33
File and volume manipulation . . . . . U 32–33
Information and drawing environment . . . . . U 32–33
Input and output . . . . . U 32–33
Primary . . . . . S 3, U 10–11
Secondary . . . . . S 3, U 10–12

Commands:
Definition . . . . . U 2
Entering . . . . . L 12, S 4
HP EGS module . . . . . S 6
Loading with LOAD . . . . . S 161
Methods . . . . . L 15
Options . . . . . L 15
Primary . . . . . S 3
Secondary . . . . . S 3
Syntax . . . . . S 1
Terminating . . . . . L 25, S 5, S 178
Terminators . . . . . L 15
Understanding . . . . . S 3
Used on screen menus . . . . . L 15
Communicating with HP EGS . . . . . L 12
*COMP . . . . . L 141
Comparing connection lists . . . . . L 266, L 271, U 182–219
Preparing to compare . . . . . U 184
Default Menu file . . . . . U 185
Compiler . . . . . M 73
Completing commands . . . . . S 5
Component:
Adding with lock angle enforcement . . . . . S 165
Constants . . . . . U 71
Copying . . . . . L 25
Copying restricted . . . . . L 45
Defining color with EQUATE . . . . . S 117
Definition . . . . . U 2
Deleting . . . . . L 31, S 90–91
Descriptors . . . . . S 2, U 18
Dynamic tracking . . . . . U 16
Filling . . . . . L 24
Hatching . . . . . S 26–28
Identifying . . . . . L 49
Moving . . . . . L 70
Moving with MOVE . . . . . S 176–177
On screen menus . . . . . L 15
Restrictors . . . . . L 15
Supported by PHOTOPLOT . . . . . U 293
Computers used with HP EGS, models 217, 220, 236, 237, 310, and 320 . . . . . I 8–9
CONCNST (Connection Lister default string constants) . . . . . U 171
Conditional macros . . . . . U 76
CONERR (Connection Lister errors) . . . . . U 167–170

Configured disc, connecting . . . . . I 15–16
Connect . . . . . L 179
Connected part, moving . . . . . L 195
Connecting:
  Passive parts . . . . . L 185
  Traces . . . . . L 259, L 278
Connection list:
  Comparing . . . . . L 266, L 271, U 182
  Creating . . . . . L 201
  Efficient processing . . . . . U 157
  Format . . . . . U 157–166
  Making . . . . . L 200, L 268
  Merging . . . . . U 173–177
  Preparing . . . . . L 200
  Utilities . . . . . U 173
Connection List Comparison:
  BACK_ANNOTATE . . . . . U 182–183
  Back annotation file . . . . . U 195, U 182–183
  Back-annotating a schematic drawing . . . . . U 206
  Comparing connection lists . . . . . U 182–219
  Example . . . . . U 207
  Message file . . . . . U 189
  Pin and gate swapping analysis . . . . . U 188
  Report file . . . . . U 196
  Types of swapping . . . . . U 194
  Understanding swap analysis . . . . . U 191
  Using the program . . . . . U 198–205
Connection Lister:
  Archiving drawings . . . . . U 145
  Associated text . . . . . U 144
  Customizing . . . . . U 171–172
  Default files . . . . . U 150–155
  Error file . . . . . U 167–170
  Menu options . . . . . U 149
  Message file . . . . . U 156, U 171–172
  Operational errors . . . . . U 167
  Output file . . . . . U 157–166
  Using . . . . . U 143–172

Connection listing:
Bus Notation . . . . . U 141
Concepts . . . . . U 133–142
Definition of Terms . . . . . U 138
Flat . . . . . U 140
Hierarchical . . . . . U 140
Implicitly stepped components . . . . . U 142
Logical rules . . . . . U 139
Physical rules . . . . . U 139
Properties . . . . . U 142
Special reference designators . . . . . U 141
Connection hierarchy . . . . . U 139
Connections, adding . . . . . L 194
*CONS . . . . . L 26
Consolidating free space on volume . . . . . M 7
Construction circles, adding . . . . . L 106
Construction lines:
Adding . . . . . L 26, L 59, L 105, L 112
Deleting . . . . . L 115
CONTINUE ON ERROR (Photoplot/Drill) . . . . . U 316
Control file:
Building . . . . . U 42
Structure . . . . . U 42
Convert:
Character code . . . . . M 20
Old parts file . . . . . U 114
Coordinates, locating with POINT . . . . . S 191
Copy . . . . . M 40, M 95
COPY* . . . . . L 25
COPY:
Chain . . . . . S 71–74
Command description . . . . . S 70–74
Datum . . . . . S 71–74
Copying:
Components . . . . . L 25
Files . . . . . M 59, M 138
Files with FILECOPY . . . . . S 124–125
Instances . . . . . L 68
Restricted components . . . . . L 45
Set . . . . . L 138
Text in text editor . . . . . M 40, M 95
View . . . . . L 133

Correcting:
Location errors, x,y . . . . . **L** 26
Pad locations . . . . . **L** 243
CPU ID Prom . . . . . **I** 75
Cpy . . . . . **M** 40, **M** 95
Creating:
Artwork . . . . . **L** 266
Assembly drawing . . . . . **L** 275
Bill of materials . . . . . **L** 214
Circuit side physical layer . . . . . **L** 276
Connection list . . . . . **L** 201
Drawing with EDIT . . . . . **S** 110–115
Ground physical layer . . . . . **L** 277
Holes . . . . . **L** 116
Library parts . . . . . **U** 79–87
Macros with DEFINE . . . . . **S** 84–89
Material list . . . . . **L** 211
Material list data . . . . . **L** 156
Parts list . . . . . **L** 159
Schematic library part . . . . . **L** 185
CS/80 tapes; backing up system on . . . . . **M** 16
CTRL BACK SPACE . . . . . **L** 6
CTRL CLR LN . . . . . **L** 6
Current:
Command . . . . . **U** 59
Working directory . . . . . **M** 11
CURSOR . . . . . **S** 76–79, **U** 16, **U** 22
Cursor:
In text editor . . . . . **M** 27, **M** 32
Moving . . . . . **L** 13
Size . . . . . **S** 5, **S** 76–78
Snapping modes . . . . . **L** 17
User-defined . . . . . **S** 76–79
Customizing:
Steps . . . . . **U** 35

I *Installing HP EGS 2.2*
L *Learning HP 2.2*
M *Managing and Editing Files with HP EGS 2.2*
S *HP EGS 2.2 Syntax Reference*
U *Understanding HP EGS 2.2*

d

*DATA . . . . . L 103, L 147
Date:
  Entering . . . . . L 8
  Setting system date . . . . . M 85
DATUM:
  Command description . . . . . S 81
  Used like COPY :D . . . . . S 81
Datum:
  Copying . . . . . S 71–74
  Dimensioning . . . . . S 19
Debugger . . . . . M 73
DECIMALS . . . . . S 82
DECIMALS (archive) . . . . . S 267
Default files:
  Connection Lister . . . . . U 150–155
  IGES Translator . . . . . U 353–358, U 366–367
  Photoplot/Drill . . . . . U 310–312
Default:
  Commands . . . . . U 32–33
  String constants . . . . . U 171
  Values, process file . . . . . U 53–54
  Volume . . . . . M 6
  Volume, specifying with PREFIX . . . . . S 192
DEFINE . . . . . S 84–89, U 62, U 81
DEFINE (archive) . . . . . S 268–269
Defining:
  Layers . . . . . U 52
  Macros with DEFINE . . . . . S 84–89, U 62
  Module . . . . . U 38
  Part geometry . . . . . L 185
  PC board library part . . . . . L 240
DEL CHR . . . . . L 5
DELETE . . . . . S 90–91
Delete . . . . . M 34, M 97
DELETE* . . . . . L 31

Deleting:
Center lines . . . . . L 145
Center lines and fillets . . . . . L 139
Components . . . . . L 31
Construction lines . . . . . L 115
Hidden lines and fillets . . . . . L 139
Hidden portions . . . . . L 137
Text in text editor . . . . . M 34, M 97
Descriptors, component . . . . . S 2
Detail layer:
Defining with EQUATE . . . . . S 117
Type . . . . . U 19
Detail mode, setting display with WINDOW . . . . . S 231
Detailing the hub . . . . . L 114
*DIAM . . . . . L 30
Diameter dimension . . . . . S 19
Diameter holes, adding . . . . . L 108
Diametral dimensions, adding . . . . . L 123
Differences between HP EGS 2.1 and 2.2 . . . . . I 99–100
Digitizer, Summagraphics . . . . . I 23–24
Digitizing:
Drawings with CALIBRATE . . . . . S 63–65
Tablet menus with SAVE . . . . . S 201
Dimension:
Extension lines, distance between . . . . . S 94–95
Tolerance values . . . . . S 18–22
Tolerance values, font size of . . . . . S 100
Dimension notes, adding . . . . . L 124, S 18–22
Dimension values:
Font size . . . . . S 96
Number of decimal places . . . . . S 92
Orientation . . . . . S 93
Radix . . . . . S 97
Scale . . . . . S 99
Slant . . . . . S 101
Units . . . . . S 102–103

Dimensioning:
Chamfer . . . . . L 127
Circle location . . . . . L 90
Diameter of a circle . . . . . L 91
Face of a block . . . . . L 91
Fillet radius . . . . . L 126
DIMENSION, ADD . . . . . S 18
*DIMN . . . . . L 103
DIM_DEC . . . . . S 92
DIM_DEC (archive) . . . . . S 270
DIM_DIR . . . . . S 93
DIM_DIR (archive) . . . . . S 271
DIM_EXTEND . . . . . S 94–95
DIM_EXTEND (archive) . . . . . S 272
DIM_FSIZE . . . . . S 96
DIM_FSIZE (archive) . . . . . S 273
DIM_RADIX . . . . . S 97
DIM_RADIX (archive) . . . . . S 274
DIM_SCALE . . . . . S 99
DIM_SCALE (archive) . . . . . S 275
DIM_TFSIZE . . . . . S 100
DIM_TFSIZE (archive) . . . . . S 276
DIM_TSLANT . . . . . S 101
DIM_TSLANT (archive) . . . . . S 277
DIM_UNITS . . . . . S 102–103
DIM_UNITS (archive) . . . . . S 278–279
Directories, listing . . . . . M 55, M 144
Disc drive, finding the drive number (unit #) . . . . . I 5
DISPDISP . . . . . L 235
DISPLAY . . . . . S 104–105
Display:
Grid, setting with GRID . . . . . S 139–141, U 14–15
Optimization layers . . . . . L 233
Option switches, setting on a video interface card . . . . . I 21–22
Setting the default with the Codeword utility . . . . . I 77
DISPPEN . . . . . L 235
DISTANCE . . . . . S 106
DlBox . . . . . L 55
DlCom . . . . . L 55
Dlete . . . . . M 34, M 97

Drawing:
Area, changing, zooming, and magnifying with WINDOW . . . . . S 231–232
Components . . . . . U 18
Creation and editing commands . . . . . U 32–33
Environment . . . . . U 52
Environments, defining with EQUATE . . . . . S 116–119
Files . . . . . U 17
Hierarchy . . . . . U 30
Hub . . . . . L 110
Initializing . . . . . L 120
Isometric rectangle . . . . . L 82
Loading with LOAD . . . . . S 161
Location, setting with WINDOW . . . . . S 231
Magnification, setting with WINDOW . . . . . S 231
Modifying with MODIFY . . . . . S 172–175
Origin placement . . . . . U 79
Plotting . . . . . L 38
Printing . . . . . L 38
Re-editing . . . . . L 250
Saving . . . . . L 36, L 45
Saving with SAVE . . . . . S 200–202
Scale . . . . . U 79
Tablet menu . . . . . U 90
*DRID . . . . . L 154
DRILL:
Description . . . . . U 279
Output . . . . . U 303–305
Overview . . . . . U 303
Virtual DRILL output . . . . . U 304–305
DRILL FORMAT (Photoplot/Drill) . . . . . U 317
DRILL ORIGIN (Photoplot/Drill) . . . . . U 303, U 319
Drill post-processor:
Codeword . . . . . U 282–286
Installing . . . . . U 280–282
Overview . . . . . U 303
Using . . . . . U 287–288

DRILL UNITS FACTOR (Photoplot/Drill) . . . . . U 303, U 323
*DSRY . . . . . L 55
DUMP . . . . . S 107
DUMP* . . . . . L 38
Duplicate link . . . . . M 133
Dynamic component tracking . . . . . U 16
Dynamic tracking, terminating with NOP . . . . . S 178

e

ECHO . . . . . S 109
Echo file . . . . . S 109
Edge connector, adding . . . . . L 251
Edge connectors . . . . . L 282, L 287
EDIF Translator:
  Default file . . . . . U 237
  Example HP EGS to HP PCDS transfer . . . . . U 269
  File format . . . . . U 253
  Handling EDIF translator errors . . . . . U 246
  HP PCDS data restrictions . . . . . U 267
  Keywords . . . . . U 255
  Map file . . . . . U 262
  Message file . . . . . U 243
  Non-EDIF restrictions . . . . . U 260
  Preparing your connection list . . . . . U 228
  Using the EDIF translator . . . . . U 231
EDIT . . . . . S 110–115
EDIT (archive) . . . . . S 280
Editing:
  Keys . . . . . U 9–10
  Message files . . . . . U 97
Editor . . . . . M 27, M 92
Editor, using to define macros . . . . . S 86
Edit_PCB . . . . . L 234
Edit_Sch . . . . . L 178
EGS/IGES Translator:
  Description . . . . . U 352–364
  Limitations . . . . . U 363–364
  Messages . . . . . U 362
  Relationships . . . . . U 351
EGS/IGES Translator default file:
  Default dimension arrowheads . . . . . U 358
  Dimension arrow . . . . . U 356
  Form parameters . . . . . U 354–356
  IGES note fonts . . . . . U 355
  Leader arrow . . . . . U 356

Memo parameters . . . . . U 354
Note and text form numbers . . . . . U 355
Note and text stroke files . . . . . U 354
Tape driver parameters . . . . . U 357
Electrical Schematic Drawing:
Description . . . . . L 1
Loading . . . . . L 181
Material listing . . . . . U 109, U 127
Module files . . . . . M 24
Module installation on a hard disc . . . . . I 41–42, I 53–54
Module installation on an SRM . . . . . I 66–67
Schematic Drawing . . . . . L 1, L 175
Electrical schematic connection lister . . . . . U 146, U 147, U 152
END_ASSOC . . . . . S 281
Enforcement of lock angle . . . . . S 164–165
Engineering Graphics files . . . . . M 21
Enter . . . . . L 23
Entering:
Commands . . . . . L 12
Data . . . . . L 5
EOC . . . . . L 25
EQUATE . . . . . S 116, U 19, U 52
EQUATE (archive) . . . . . S 282–284
Errors:
Connection Lister . . . . . U 167–170
IGES Translator . . . . . U 362, U 370–372
Photoplot/Drill . . . . . U 336–339
Run-time errors . . . . . M 171
EVALUATE . . . . . S 120–122, U 16
Excellon drill files . . . . . U 279, U 303
eXchange . . . . . M 36, M 121
eXecute . . . . . M 89
Executing a code file . . . . . M 89
EXIT . . . . . S 123
EXIT (archive) . . . . . S 285
EXIT, use of during an instance-in-context EDIT . . . . . S 114
Expressions, creating with DEFINE . . . . . S 84–89
Extended directory . . . . . M 135
External monitor, setting default with codeword utility . . . . . I 77

---

f

f2 key . . . . . **L** 6
f3 key . . . . . **L** 6
File:
  Changing file name . . . . . **M** 60, **M** 131
  Changing file type . . . . . **M** 61, **M** 157
  Concepts . . . . . **M** 1
  Copying . . . . . **M** 59, **M** 138
  Copying with FILECOPY . . . . . **S** 124–125
  Drawing . . . . . **U** 17
  Moving . . . . . **M** 59, **M** 138
  Names . . . . . **M** 2
  Operations . . . . . **L** 79
  Plotting with PLOT . . . . . **S** 186
  Removing with REMOVE . . . . . **S** 194–195
  Saving with SAVE . . . . . **S** 200–202
  Specifications . . . . . **M** 1
  Suffixes . . . . . **S** 69
  Table, displaying or changing system file table . . . . . **M** 87
File and volume manipulation commands . . . . . **U** 32–33
File management:
  Breaking up large files . . . . . **M** 18
  Re-creating large files . . . . . **M** 18
FILECOPY . . . . . **S** 124–125, **L** 80
Filecopy . . . . . **M** 59, **M** 138
Filer . . . . . **M** 125
$FILES . . . . . **S** 286
FILL . . . . . **L** 24, **L** 27, **L** 74
FILL (used with PLOT*) . . . . . **L** 38
Fill modes with PHOTOPLOT . . . . . **U** 294
FILL WIDTH (Photoplot/Drill) . . . . . **U** 317
Fillet . . . . . **S** 11
Fillet radius, dimensioning . . . . . **L** 126
Filling in Items . . . . . **L** 24
Filling paragraphs in text editor . . . . . **M** 46, **M** 106
*FILT . . . . . **L** 40
Find . . . . . **M** 42, **M** 99

Flange, adding . . . . . L 120
Flange drawing:
Annotating . . . . . L 120
Creating . . . . . L 97
Flange part:
Adding . . . . . L 132
Adding items . . . . . L 148
Flat connection listing . . . . . U 140
Flexible discs:
Initializing . . . . . M 13
Inserting and removing . . . . . I 33
Use of . . . . . I 33–34
*FONT . . . . . L 74, L 148
Font size:
Changing with MODIFY . . . . . S 173
Of notes and text . . . . . S 53-55
Setting with FSIZE . . . . . S 126
FORMAT (Material listing) . . . . . U 118–122
Fractional resolution of dimension values . . . . . S 103
*FREE . . . . . L 28
FREE:
ADD LINE . . . . . S 32–37
ADD POLYGON . . . . . S 45
Front view dimensions, adding . . . . . L 125
*Front30 . . . . . L 82
FSIZE . . . . . S 126
FSIZE (archive) . . . . . S 287
Functions, calculating with EVALUATE . . . . . S 121

g

GATHER . . . . . S 127–132
GATHER* . . . . . L 54
GATHER* options . . . . . L 55
gedit_ms.ASC file, entering codeword in . . . . . I 77–79
GENERATE . . . . . L 77, S 134–136, U 280, U 352

Generate:
Files . . . . . **U** 280
Post-processed material list . . . . . **U** 124, **U** 130, **U** 134
Rat's nest . . . . . **U** 178
Geometry . . . . . **L** 179
Geometry mode . . . . . **L** 101
Gerber photoplotter . . . . . **U** 279
Get . . . . . **M** 141
$GR . . . . . **L** 17
Graphical EDIT . . . . . **S** 111–112
Graphics and alpha screens . . . . . **L** 16
Graphics:
Input device . . . . . **L** 13, **U** 3
Screen . . . . . **L** 16, **U** 7
Graphics dump . . . . . **S** 107
Graphics editor:
Description . . . . . **U** 1–2
Files . . . . . **M** 21
Graphics tablet:
Loading with LOAD . . . . . **S** 160–161
Saving tablet menus with SAVE TMENU . . . . . **S** 201
Setting the bus address . . . . . **I** 19–20
Setting the default with codeword utility . . . . . **I** 77
Toggling on and off with TMENU . . . . . **S** 221
Using to digitize a drawing with CALIBRATE . . . . . **S** 63–65
GRID . . . . . **S** 138–141, **U** 22
GRID (archive) . . . . . **S** 288
:GRID snapping mode . . . . . **S** 4, **U** 15
Grid:
Description of system . . . . . **U** 12–15
Display . . . . . **S** 139–141, **U** 14–15
System . . . . . **L** 17, **S** 226–227, **U** 12–13
User . . . . . **L** 18, **U** 14–15
Using while building library parts . . . . . **U** 79
GROUP . . . . . **S** 142–143

h

:H, used to define an unhatched region . . . . . **S** 25
Hard disc:
  Copying files . . . . . **I** 91
  Emptying . . . . . **I** 91–92
  Finding contents of . . . . . **I** 89–90
  Initializing . . . . . **I** 85–88
  Listing volume contents . . . . . **I** 89–90
  Removing files . . . . . **I** 92
  Setting the bus address . . . . . **I** 18
  Setting up an HP 9133 or 9134 . . . . . **I** 17
  Setting up an HP 9153 . . . . . **I** 15–17
Hardware:
  Connecting for Installation . . . . . **I** 11–26
  For HP EGS . . . . . **I** 7–10
HATCH, ADD . . . . . **S** 24–28
Hatching:
  Changing style with MODIFY . . . . . **S** 173
  Options . . . . . **L** 41
  With different snapping modes . . . . . **S** 27–28
HB-IB cables, connecting . . . . . **I** 12
HELP . . . . . **S** 145–147
HELP (Photoplot/Drill) . . . . . **U** 318
Help messages . . . . . **U** 101
HELP* . . . . . **L** 34
Hidden lines and fillets, deleting . . . . . **L** 139
Hidden portions, deleting . . . . . **L** 137
Hierarchical:
  Connection listing . . . . . **U** 140
  Directory structure . . . . . **M** 8
Hierarchy:
  Connection . . . . . **U** 139
  Drawing . . . . . **U** 30
  Logical . . . . . **U** 31
*HILO . . . . . **L** 91
*HOLE . . . . . **L** 41

I *Installing HP EGS 2.2*
L *Learning HP 2.2*
M *Managing and Editing Files with HP EGS 2.2*
S *HP EGS 2.2 Syntax Reference*
U *Understanding HP EGS 2.2*

Hole:
Adding . . . . . **L** 88
Creating . . . . . **L** 116
HORizontal dimension values with DIM_DIR . . . . . **S** 93
Horizontal line dimension . . . . . **S** 19
Hpegs . . . . . **M** 74
HP EGS 2.1, differences between 2.1 and 2.2 . . . . . **I** 99–100
HP EGS 2.2:
Additional enhancements since HP EGS 2.1 . . . . . **I** 100
File naming conventions . . . . . **U** 36
Hardware differences since HP EGS 2.1 . . . . . **I** 100
Manager . . . . . **U** 50
Option 085, connecting the configured disc . . . . . **I** 15–16
Software installation . . . . . **I** 27–84
Stroke files . . . . . **U** 97
System differences since HP EGS 2.1 . . . . . **I** 99–100
User interface differences since HP EGS 2.1 . . . . . **I** 100
HP TechWriter, readying drawings for . . . . . **L** 38
HPTW . . . . . **L** 38
HP 9133 or 9134 disc, setting up . . . . . **I** 17
HP 9153 disc, setting up . . . . . **I** 15–17
*HrVr . . . . . **L** 90
HrVrt . . . . . **L** 59
Hub:
Detailing . . . . . **L** 114
Drawing . . . . . **L** 110

i

I-C DIP parts . . . . . **L** 287
ID module, linking to an HP EGS codeword . . . . . **I** 72–79
IDENTIFY . . . . . **S** 149–150
Identifying components . . . . . **L** 49
IFELSE function in macros . . . . . **U** 76–77
IGES Translator:
Codeword installation . . . . . **U** 345–349
Copious data . . . . . **U** 351
Default File . . . . . **U** 366–367
Directory entry section . . . . . **U** 350
Files . . . . . **M** 27
General note . . . . . **U** 351

Global section . . . . . U 350
Limitations . . . . . U 372
Local installation . . . . . U 343
Messages . . . . . U 370–372
Parameter data section . . . . . U 350
Retrieving drawings . . . . . U 365
SRM installation . . . . . U 344
Start section . . . . . U 350
Subfigure instance . . . . . U 351
Terminate section . . . . . U 351
IGNORE identifier . . . . . U 141, U 172
Implicitly stepped components . . . . . U 142
Implicitly stepped components, smashing with SMASH . . . . . S 207
$IN . . . . . L 17
in>mm . . . . . L 102
INAME . . . . . S 151
INAME (archive) . . . . . S 289
*INBT . . . . . L 137
Indexing message files . . . . . U 98
Information and drawing environment commands . . . . . U 32–33
Initialization:
Explanation of . . . . . I 85–88
Peforming with the Zero command . . . . . I 93–94
Initialize . . . . . M 75
Initializing:
Drawing . . . . . L 120, L 132
File system . . . . . M 75
Flexible discs . . . . . M 13
INLine dimension values with DIM_DIR . . . . . S 93
Inner layer:
Pads . . . . . L 280, L 286
Planes . . . . . L 231
INPUT:
Call up start file . . . . . U 48
Command . . . . . L 78, S 152–153
In control files . . . . . U 46
(Photoplot/Drill) . . . . . U 318

Input and output commands . . . . . **U** 32–33
Input device:
Described by locator tag . . . . . **S** 163
Graphics . . . . . **L** 13
Input file:
Creating from a GENERATE file . . . . . **S** 152
Pausing with PAUSE . . . . . **S** 181
INS CHR . . . . . **L** 5
Insert . . . . . **M** 29, **M** 102
Installation:
EWME on a hard disc . . . . . **I** 44–45, **I** 56–57
EWME on an SRM . . . . . **I** 70–71
EWPCB on a hard disc . . . . . **I** 43–44, **I** 55–56
EWPCB on an SRM . . . . . **I** 68–69
EWRC on a hard disc . . . . . **I** 39–40, **I** 51–52
EWRC on an SRM . . . . . **I** 64–65
EWSCH on a hard disc . . . . . **I** 41–42, **I** 53–54
EWSCH on an SRM . . . . . **I** 66–67
EWSYS on a hard disc . . . . . **I** 38, **I** 49–50
EWSYS on an SRM . . . . . **I** 62–63
Hardware needed for . . . . . **I** 7–9
HP EGS codeword . . . . . **I** 72–80
HP EGS on an existing hard disc . . . . . **I** 46–57
HP EGS on a new hard disc . . . . . **I** 35–45
HP EGS on an SRM . . . . . **I** 58–71
HP EGS on top of Pascal 3.12 . . . . . **I** 101–103
HP EGS Option 085 . . . . . **I** 27
HP EGS Options 042 and 044 . . . . . **I** 28–84
IGES Translator . . . . . **U** 343–349
Pascal 3.12 alongside HP EGS 2.2 . . . . . **I** 103
Photoplot/Drill . . . . . **U** 280–286
Terms used . . . . . **I** 5
Verifying . . . . . **I** 81–83
Instance-in-Context EDIT . . . . . **S** 112–115
Instance-oriented connection list . . . . . **U** 157–166
Instance:
ADD . . . . . **S** 30–31
Adding . . . . . **L** 63
Copying . . . . . **L** 68
Creating with WRAP . . . . . **S** 234–237
Description of . . . . . **U** 28–31, **U** 138
Editing with EDIT . . . . . **S** 111–112

Saving with SAVE . . . . . S 200–202
Smashing . . . . . L 69
Used to create library parts . . . . . U 80
Viewing . . . . . L 184
Interconnect:
Layer, defining . . . . . S 117
Layer type . . . . . U 19
Mode, setting display . . . . . S 231
Interface card, video . . . . . I 21–22
Internal units per user unit, used with UNITS . . . . . S 227
Intersect . . . . . S 4
:INTERSECT snapping mode . . . . . U 15
Isometric:
Area . . . . . S 57
Grid . . . . . S 139–141
Grid, cursor in . . . . . S 78
Rectangle . . . . . L 82, L 84
Isometric drawing:
Archive files . . . . . U 24–28
Converting orthogonal views . . . . . U 24–28
Isometric conversions . . . . . U 23–24
Procedures . . . . . U 21–28
Views (front, top, right) . . . . . U 22
ITEM . . . . . L 103

j

Jump . . . . . M 105
JUNCTION . . . . . L 180
*JUST . . . . . L 74
Justification value for notes and text . . . . . S 49, S 51

k

Keepout area, adding . . . . . L 243
Keepout layer, master . . . . . L 232
Keyboard cables, connecting . . . . . I 13–14
Krunch . . . . . M 142

---

l

Large file management . . . . . **M** 18
Largest space on a disc . . . . . **S** 67
LaserJet printer, configuring for HP EGS . . . . . **I** 25, **I** 105–107
Layer:
  Definitions . . . . . **L** 52, **L** 99, **L** 227, **U** 52
  Description . . . . . **U** 18–19
  Detail . . . . . **U** 19
  Interconnect . . . . . **U** 19
  Label . . . . . **U** 52
  Number . . . . . **U** 52
  Symbolic . . . . . **U** 19
  Using while creating library parts . . . . . **U** 80
Layers, defining with EQUATE . . . . . **S** 117
*LEAD . . . . . **L** 26, **L** 94
Leader line option of ADD LINE . . . . . **S** 33
Leader lines, adding . . . . . **L** 151
LEVEL . . . . . **S** 155, **U** 53
LEVEL (archive) . . . . . **S** 290
LEVEL* . . . . . **L** 179, **L** 234
Levels, nesting . . . . . **L** 66
Librarian . . . . . **M** 73
Library parts:
  Creating a schematic . . . . . **L** 185, **U** 79–86
  Description . . . . . **U** 17
  Drill holes (Photoplot/Drill) . . . . . **U** 334
  Flash apertures (Photoplot/Drill) . . . . . **U** 330–331
  Ignore and no fill tags (Photoplot/Drill) . . . . . **U** 333
  Macro instances . . . . . **U** 80–85
  PC board . . . . . **L** 240, **L** 283
  Polygon, hatch, rectangle fill aperture (Photoplot/Drill) . . . . . **U** 331–333
  Schematic . . . . . **L** 220
  Special considerations . . . . . **U** 79–80
  Text/note draw aperture (Photoplot/Drill) . . . . . **U** 333
LINE . . . . . **L** 28, **L** 168

Line:
Adding multivertex . . . . . L 43
Construction . . . . . L 26
Defining type . . . . . U 52
Deleting . . . . . L 145
Dimension . . . . . S 19
Replacing portion with STRETCH . . . . . S 218
Stretching . . . . . L 142
Style . . . . . S 118
Type . . . . . S 117
LINE, ADD . . . . . S 32–37
Linear dimensions, adding . . . . . L 168
LIST . . . . . S 156–158, L 78
List:
Directory . . . . . M 55, M 144
Parts file . . . . . U 116
Post-processed material list . . . . . U 126
LIST MATERIAL:
Command . . . . . S 157
Output file . . . . . U 110, U 111
LIST_MAT . . . . . L 103, L 179, L 235
lkNEW . . . . . L 59
lkON/lkOFF . . . . . L 59
*LMEN . . . . . L 262
LNTYP . . . . . L 44
*LNUM . . . . . L 262
LOAD . . . . . S 159–162, U 31, U 48, U 52, U 62, U 94, U 101
Loading:
Electrical Schematic Drawing . . . . . L 181
Engineering Graphics . . . . . L 11
Mechanical Drafting . . . . . L 104
Printed Circuit Board Layout . . . . . L 237
Programs . . . . . S 161
Tablet Menus . . . . . S 161

Local mass storage:
Checking readability . . . . . M 67, M 130
Compacting files . . . . . M 67, M 142
Creating an empty directory . . . . . M 68, M 163
Identifying unreliable disc sectors . . . . . M 130
Location errors, correcting . . . . . L 26
LOCATOR . . . . . S 163
Locator, setting with the Codeword utility . . . . . I 77
LOCK . . . . . S 164–165
LOCK (archive) . . . . . S 291
Lock angle warning . . . . . S 164–165
Logical levels . . . . . L 192, L 248
Logical operators, calculating with EVALUATE . . . . . S 120–121
Logical ports, adding . . . . . L 189
Logical:
Axis . . . . . U 21–24
Connection rules . . . . . U 139
Level . . . . . U 31, U 80
LOG_PORT . . . . . L 180

m

Macro:
Building or modifying a file . . . . . U 62
Conditional . . . . . U 76–77
Creating with DEFINE . . . . . S 86–89
Description . . . . . U 62
How to modify existing information . . . . . U 68
How to save values and set modes . . . . . U 68
HP EGS module . . . . . S 6
Recursive . . . . . U 67
Short vs. long . . . . . U 64
Substituting for command . . . . . U 65
Syntax rules . . . . . U 63
Temporary . . . . . U 66
Macro constants, system-defined . . . . . U 69–75
Macro file:
For each module . . . . . L 101, L 178, L 233
Listing file contents with LIST . . . . . S 157
Loading file with LOAD . . . . . S 160
Saving file with SAVE . . . . . S 200–202

Macro instance:
Archiving as a graphical instance . . . . . **S** 54
Creating with DEFINE . . . . . **S** 86–89, **U** 79–87
Description . . . . . **U** 17, **U** 79–87
Editing with EDIT . . . . . **S** 112–115, **U** 79
Library parts . . . . . **U** 17, **U** 29, **U** 80–85
Mag/paper tape utility . . . . . **M** 19
*MAKE . . . . . **L** 55
Make . . . . . **M** 146
Make_Prt . . . . . **L** 178, **L** 234
Making connection lists . . . . . **L** 200, **L** 268
Making material lists . . . . . **L** 200
Manager menu, explanation of . . . . . **I** 82–83
Manufacture and assembly layers . . . . . **L** 230
Margin . . . . . **M** 46, **M** 106
MARKER, ADD . . . . . **S** 39
Mass storage, IGES Translator requirements . . . . . **U** 364, **U** 373
Master keepout layer . . . . . **L** 232
Material list data, creating . . . . . **L** 156
Material list:
Adding . . . . . **L** 163
Creating . . . . . **L** 200, **L** 211
Description . . . . . **U** 109
Error checking . . . . . **U** 127
File, preparing with LIST . . . . . **S** 157
Listing . . . . . **U** 126
Of an assembly drawing . . . . . **L** 156
Output format . . . . . **U** 117–124
Parts file editor . . . . . **U** 111
Mathematical:
Calculations with EVALUATE . . . . . **S** 120
Expressions . . . . . **U** 16
Expressions, creating with DEFINE . . . . . **S** 84–89
Matrix, creating radial matrix with STEP . . . . . **S** 208–211
*MDFY . . . . . **L** 55

Mechanical Drafting:
Drawings . . . . . L 1
Module . . . . . L 97
Module file . . . . . M 25
Module, installation on a hard disc . . . . . I 44–45, I 56–57
Module, installation on an SRM . . . . . I 70–71
Module loading . . . . . L 104
Post-processor . . . . . U 109, U 131
Tablet menu . . . . . L 173
Mechanical library parts . . . . . L 169
MEMORY . . . . . S 167
Memory:
Requirements for HP EGS . . . . . I 7–9
Used by loading commands . . . . . S 161
Memvol . . . . . M 76
MENU . . . . . S 168–170, U 57
Menu file, loading with LOAD . . . . . S 160
Menu:
Height . . . . . U 59
Horizontal or vertical . . . . . U 59
Items . . . . . L 16
Loading with LOAD . . . . . S 159–161
Manager . . . . . I 82–83
Modifying and moving screen menus with MENU . . . . . S 168–170
Row . . . . . U 58
Saving tablet menus with SAVE TMENU . . . . . S 201
Toggling on tablet menus with TMENU . . . . . S 221
Width . . . . . U 59
Merging connection lists . . . . . U 173–177
Message file:
Customizing (Connection Lister) . . . . . U 171–172
Customizing (Photoplot/Drill) . . . . . U 326–329
Editing and indexing . . . . . U 98
Loading with LOAD HELP . . . . . S 160
Modifying . . . . . U 97–101
Message tags . . . . . U 97
*MIR . . . . . L 56
Mirroring of notes and text . . . . . S 49
mm>in . . . . . L 102
Model number, finding at boot-up . . . . . I 75

Modes . . . . . L 15
MODIFY . . . . . S 172–175
Modifying:
  Assembly . . . . . L 144
  Drawing or instance . . . . . S 110–115
  Macro file . . . . . U 62
  Message file . . . . . U 97
  Multiple components . . . . . L 54
  Notes and text . . . . . L 198, L 253
  Parts file . . . . . U 115
  Parts screen menus . . . . . L 295
  Process file . . . . . U 52
  Tablet menu . . . . . L 297, U 88
  Window . . . . . L 51
MOD_NOTE* . . . . . L 177, L 232
Module, customized . . . . . U 106
Module macros . . . . . S 6
Monitor, setting default with codeword utility . . . . . I 77
Mouse . . . . . L 13
MOVE . . . . . S 176–177
*MOVE . . . . . L 56
MOVE* . . . . . L 70
Moving:
  Components . . . . . L 70
  Connected part . . . . . L 195
  Cursor . . . . . L 13
  Cursor in text editor . . . . . M 32
  Files . . . . . M 59, M 138
  Parts list . . . . . L 165
  Text in text editor . . . . . M 40, M 95

n

Naming conventions . . . . . U 36–37
Naming nets . . . . . L 197
Nesting level:
  Description . . . . . L 66, U 28–31, U 80
  Setting with WINDOW . . . . . S 231
Net . . . . . U 138

Net element . . . . . U 138
Net name . . . . . U 138
Net-oriented connection list . . . . . U 157–162
Nets, naming . . . . . L 197
Netting level . . . . . U 140
NET_NAME . . . . . L 180, L 235
New . . . . . M 149
Newsysvol . . . . . M 78
NO FILL USE (Photoplot/Drill) . . . . . U 318
*NOLK . . . . . L 262
NOP . . . . . S 178
Note:
  ADD . . . . . S 52–56
  Adding . . . . . L 94
  Changing with MODIFY . . . . . S 173
  Setting font size with FSIZE . . . . . S 126
Note string, modifying . . . . . L 198, L 253
Note stroke table, loading with LOAD NSTROKE . . . . . S 160
NOTE USE:
  (Photoplot/Drill) . . . . . U 318
  Specifying aperture . . . . . U 307
NO_FILL identifier (Photoplot/Drill) . . . . . U 307
*NUMB . . . . . L 148
Number of decimal places . . . . . S 82
Numeric operators, calculating with EVALUATE . . . . . S 120
Nut and washer items, adding . . . . . L 149

O

Operating system files . . . . . M 21
Operators, calculating with EVALUATE . . . . . S 120–121
Options, definition . . . . . U 2
Orientation of grid with GRID command . . . . . S 140
Origin placement . . . . . U 79
*ORTHOG . . . . . L 82
Orthogonal axis . . . . . U 21–24
Outline . . . . . L 179
Outline, adding . . . . . L 197
Outlining a part . . . . . L 242
OVAL, ADD . . . . . S 40–42

p

$P . . . . . L 17
PACK . . . . . L 80, S 179
Pad locations, adding . . . . . L 240
Pad, defining pad one . . . . . L 244
Pad set, adding . . . . . L 240
PAGE . . . . . S 180
Page . . . . . M 107
Panning with WINDOW . . . . . S 232
*PAR . . . . . L 28
Parameter:
  Modifying with EDIT . . . . . S 115
  Values of macro instances . . . . . S 86
Part:
  Aligning . . . . . L 187, L 245
  Geometry, defining . . . . . L 185
  Information, collecting . . . . . L 147
  Modifying screen menus . . . . . L 295
  Name . . . . . U 138
  Outlining . . . . . L 242
  Type, adding . . . . . L 188, L 245
  Verification . . . . . L 191, L 247
Parts file:
  Changing . . . . . L 156
  Editor . . . . . U 111
  Listing . . . . . U 116
  Modifying . . . . . U 115
  Selecting . . . . . U 111
  Structure . . . . . U 111
Parts list:
  Creating . . . . . L 159
  Moving . . . . . L 165
  Stepping border . . . . . L 166
PART_TYP . . . . . L 180, L 235
Pascal 3.12:
  Installing alongside HP EGS 2.2 . . . . . I 103
  Installing HP EGS on top of . . . . . I 101–102

Pascal command line . . . . . M 71
Passive parts, connecting . . . . . L 185
Passwords, defining . . . . . M 127
PAUSE . . . . . S 181
PAUSE ON ERROR (Photoplot/Drill) . . . . . U 316
PC board . . . . . U 178, U 182
Peripherals:
  Bus addresses of . . . . . I 18–20
  Select codes of . . . . . I 18–19
Permanent . . . . . M 79
*PERP . . . . . L 28
Photoplot and Drill post-processor files . . . . . M 27
PHOTOPLOT FORMAT (Photoplot/Drill) . . . . . U 317
PHOTOPLOT ORIGIN (Photoplot/Drill) . . . . . U 319
PHOTOPLOT ORIGIN, using . . . . . U 295
PHOTOPLOT:
  Description . . . . . U 279
  Output . . . . . U 291
Photoplot post-processor:
  Codeword . . . . . U 282–286
  Installing . . . . . U 280–281
  Output . . . . . U 291–302
  Overview . . . . . U 289–290
  Using . . . . . U 287–290
PHOTOPLOT UNITS FACTOR (Photoplot/Drill) . . . . . U 295, U 323
PHOTOPLOT/DRILL Commands . . . . . U 312–325
Physical board layers . . . . . L 229
Physical connection rules . . . . . U 139
Physical layer:
  Circuit side . . . . . L 276
  Ground . . . . . L 277
Physical ports, adding . . . . . L 190, L 246
PHY_PORT . . . . . L 180, L 235
*PLMI . . . . . L 91
PLOT . . . . . S 182–190
PLOT* . . . . . L 38

Plotter:
Pen number . . . . . U 52
Setting the bus address . . . . . I 19
Plotting:
Drawing . . . . . L 38
Methods . . . . . L 38
Tablet menu . . . . . L 291, U 92–93
To a file . . . . . S 186
*PNTS (*3PNTS) . . . . . L 28
POINT . . . . . S 191
*POL2 . . . . . L 103
Polar copy, creating with STEP . . . . . S 207
POLYGON FILL USE:
(Photoplot/Drill) . . . . . U 319
Using tagged associated text . . . . . U 306
Polygon:
ADD . . . . . S 44–45
Adding . . . . . L 32
Replacing portion with STRETCH . . . . . S 216
Port:
Definition . . . . . U 138
Number, specifying with SHOW . . . . . S 206
On a part . . . . . U 138, U 142
Position number of screen menus . . . . . S 168
Post-processor preparation . . . . . L 254
Post-processors . . . . . U 80
Power and ground layers . . . . . L 233
Power-up display . . . . . I 26
PREFIX . . . . . L 80, S 192
PREFIX (Photoplot/Drill) . . . . . U 320
Prefix . . . . . M 54, M 150
Prefix volume . . . . . L 7, M 6
Primary pads . . . . . L 285
Primitive pads . . . . . L 284
Primitive:
Adding . . . . . L 23
Description . . . . . L 23, S 4

:PRIMITIVE snapping mode . . . . . U 15
Printed circuit board connection lister . . . . . U 147, U 150
Printed Circuit Board Layout:
  Description . . . . . L 1
  Library parts . . . . . L 283
  Loading . . . . . L 237
  Module . . . . . L 175, L 225
  Tablet menu . . . . . L 289
Printer, setting up a LaserJet . . . . . I 25, I 105–107
Printing a drawing . . . . . L 38
Printing a text file . . . . . M 47
Process file:
  Archiving with a drawing . . . . . S 54–55
  Building or modifying . . . . . U 52
  Description . . . . . L 99, L 177, L 226
  Layers . . . . . L 99, L 178, L 229
  Listing contents with LIST . . . . . S 157
  Loading with LOAD . . . . . S 160
  Placing at the beginning of a GENERATE file . . . . . S 135
  Saving with SAVE . . . . . S 200–202
  Setting resolution with RESOLUTION . . . . . S 196–197
Product number, finding at boot-up . . . . . I 75
Program:
  Loading with LOAD . . . . . S 160
  User-written . . . . . U 46
Properties . . . . . U 142
Proportionally spaced characters . . . . . U 103

q

*QTY . . . . . L 148
Quit . . . . . M 108, M 152
QUIT . . . . . S 193
QUIT (Photoplot/Drill) . . . . . U 325

r

$R . . . . . L 17
*R&C . . . . . L 30, L 88
Radial dimensions, adding . . . . . L 123
Radial matrix, creating with STEP . . . . . S 208–211
Radius dimension . . . . . S 19
*RADs . . . . . L 102
RAM, seeing the amount of at boot-up . . . . . I 26
Rat's nest:
  Generating . . . . . U 178
  Inputting traces . . . . . L 258
*RATO . . . . . L 102
:RAWPOINT snapping mode . . . . . S 4, U 15
Re-configuring HP EGS menus . . . . . M 20
Re-editing a drawing . . . . . L 250
Re-routing traces . . . . . L 260, L 262
RECALL . . . . . L 5
*RECT . . . . . L 103
Rectangle:
  ADD . . . . . S 46–47
  Replacing portion with STRETCH . . . . . S 217
Recursive macros . . . . . U 67
Redefining a pad . . . . . L 244
Reference designator:
  Adding . . . . . L 188, L 245
  Description . . . . . U 138, U 141
Reference library parts . . . . . L 283
Refreshing text editor screen . . . . . M 31, M 120
REF_DESI . . . . . L 180, L 235
Relational operators, calculating with EVALUATE . . . . . S 121
REMOVE . . . . . S 194–195, L 62, L 80
Remove . . . . . M 153
Removing:
  Construction geometry . . . . . L 118
  Files . . . . . L 62, M 61, M 153

Replace . . . . . **M** 37, **M** 111
Replacing strings in text editor . . . . . **M** 37, **M** 111
RESET (Photoplot/Drill) . . . . . **U** 320
RESOLUTION . . . . . **S** 196–197
RESOLUTION (archive) . . . . . **S** 292–293
RESOLUTION (Photoplot/Drill) . . . . . **U** 320
Resolution:
  Arc . . . . . **S** 11
  Circle . . . . . **S** 15
  Freehand line . . . . . **S** 34–37
  Oval . . . . . **S** 41
Restarting last program or subsystem . . . . . **M** 84
Restricted components, copying . . . . . **L** 45
RETRIEVE . . . . . **L** 77, **S** 196
REVISION . . . . . **S** 197
*Right30 . . . . . **L** 82
Root directory . . . . . **M** 8, **M** 9
*ROT . . . . . **L** 56, **L** 74
Rotating components with MODIFY . . . . . **S** 173
Rotation of notes and text . . . . . **S** 49
ROUTE* . . . . . **L** 179, **L** 234, **L** 262
Rplace . . . . . **M** 37, **M** 111
Rule Check post-processor files . . . . . **M** 26
Run . . . . . **M** 80
Run time:
  Estimation for IGES Translator . . . . . **U** 364, **U** 373
  Parameters . . . . . **U** 93–94

S

Save . . . . . **M** 155
SAVE . . . . . **S** 200–202, **U** 52, **U** 62
SAVE (archive) . . . . . **S** 294
Saving a drawing . . . . . **L** 36, **L** 45, **U** 52, **U** 62
*SCAL . . . . . **L** 56
SCALE . . . . . **L** 38
Scale factor, changing with MODIFY . . . . . **S** 173
Schematic:
  Library parts . . . . . **L** 220
  Tablet menu . . . . . **L** 224
Schematics . . . . . **U** 173
Scientific functions, calculating with EVALUATE . . . . . **S** 121

Screen dump:
Initiating with PAGE . . . . . S 180
With PLOT . . . . . S 185
Screen menu:
Building . . . . . U 57
Description . . . . . L 15, U 7
File structure . . . . . U 58
Modifying with MENU . . . . . S 168–170
Saving with SAVE . . . . . S 200–202
SEARCH . . . . . S 204–205, U 50
Search file:
Building . . . . . U 50–51
Listing contents with LIST . . . . . S 157
Loading with LOAD . . . . . S 160
Saving with SAVE . . . . . S 200–202
Searching for patterns in text editor . . . . . M 42, M 99
Secondary commands . . . . . U 10–12
Select . . . . . L 23
Select codes, setting on a peripheral . . . . . I 18–20
Selecting apertures with PHOTOPLOT . . . . . U 296–302
Selecting menu items . . . . . L 16
*SEMI . . . . . L 40
Semicircle and tangent lines, adding . . . . . L 106
Serial number, finding with the Codeword utility . . . . . I 75
Set . . . . . M 43, M 115
Shape of notes and text . . . . . S 51
Shape, stretching . . . . . L 32
Shared Resource Manager, see SRM
shON/shOFF . . . . . L 44
Short macros vs. long macros . . . . . U 64
SHOW . . . . . S 206, U 5
SHOW (archive) . . . . . S 295
SHOW* . . . . . L 44, L 53
Slant of notes and text . . . . . S 49
*SLNT . . . . . L 74, L 148
SMASH . . . . . S 207
SMASH identifier . . . . . U 141, U 172
Smashing an instance . . . . . L 69

Smashing level, connection listing . . . . . **U** 140
Snapping mode:
Changing during command . . . . . **S** 4
Definition . . . . . **L** 17
Example of changing during command . . . . . **S** 26
:GRID . . . . . **S** 4, **U** 15
:INTERSECT . . . . . **S** 4, **U** 15
:PRIMITIVE . . . . . **S** 4, **U** 15
:RAWPOINT . . . . . **S** 4, **U** 15
Selecting . . . . . **S** 4
Selecting points while changing . . . . . **S** 4
:VERTEX . . . . . **S** 4, **U** 15
*SNGL . . . . . **L** 91
Solder resist, adding . . . . . **L** 229
SORT APERTURE (Photoplot/Drill) . . . . . **U** 321
SORT DRILL (Photoplot/Drill) . . . . . **U** 322
SORT PROXIMITY (Photoplot/Drill) . . . . . **U** 321
SOURCE (Photoplot/Drill) . . . . . **U** 322
Source voltage input, adding . . . . . **L** 196
*SPAC . . . . . **L** 41
Space planning module . . . . . **U** 39–107
Special reference designators . . . . . **U** 141
SPOOL . . . . . **L** 38
sPT=1/sPT=2 . . . . . **L** 44
SRM:
Changing the working directory . . . . . **M** 69, **M** 150
Compared to local disc notation . . . . . **U** 37
Creating directories . . . . . **I** 98
Directory structure . . . . . **I** 95–96
Explanation . . . . . **M** 8
File organization . . . . . **M** 8
Listing directories . . . . . **I** 97
Making files and directories . . . . . **M** 69, **M** 146
Making new pointers to a file . . . . . **M** 69, **M** 133
Pathnames . . . . . **M** 9
Units and volumes . . . . . **M** 10
Start file:
Building . . . . . **U** 48
With HP EGS Graphics Editor . . . . . **U** 49

STEP . . . . . S 208–211
Step specification:
  Adding with ADD . . . . . S 7–8
  Example of . . . . . S 47
Stepping parts list border . . . . . L 167
Stick characters (note) . . . . . S 48–52
STOP key . . . . . L 6
Stopping an operation . . . . . L 79
Stream . . . . . M 81
Stream file, executing . . . . . M 81
Stream files . . . . . L 78
Streaming files with INPUT . . . . . S 152
STRETCH . . . . . S 212–220
STRETCH* . . . . . L 260
Stretching:
  Center line . . . . . L 142
  Re-routed trace . . . . . L 263
  Shape . . . . . L 32
String functions, calculating with EVALUATE . . . . . S 121
Stroke tables:
  Creating . . . . . U 102
  For notes and text . . . . . S 51
  Format . . . . . U 102
  Loading with LOAD . . . . . S 160
  Troubleshooting . . . . . U 105
  Used in connection listing . . . . . U 156
*STYL . . . . . L 41
Style, hatching . . . . . S 25
Suffixes of files . . . . . S 69
Suffixes of HP EGS files . . . . . U 36–37
Summagraphics digitizer . . . . . I 23–24
Surface finish symbol, adding . . . . . L 128
Symbolic layer:
  Defining with EQUATE . . . . . S 117
  Type . . . . . U 19
Symbolic mode, setting display with WINDOW . . . . . S 233
Syntax diagrams, interpreting . . . . . S 1
SYSTEM directory, contents of . . . . . I 96

System volume:
Definition . . . . . M 6
Designating new system volume . . . . . M 78
System-defined macro constants . . . . . U 69–75
System:
Constants . . . . . U 69–71
Cursor . . . . . S 76–79
Grid . . . . . L 17, U 12–13
Grid points . . . . . U 13–14
Grid points, assigning distance with UNITS . . . . . S 226–227
SYSTEMS directory, content of . . . . . I 96

t

Tablet menu:
Activating . . . . . L 291, U 93
Building . . . . . U 88–90
Description . . . . . U 7
Digitizing with . . . . . S 64–65
File . . . . . U 93–94
How the system interprets menu items . . . . . U 94
Loading with LOAD TMENU . . . . . S 160
Mechanical Drafting . . . . . L 173
Modifying . . . . . L 297
PC Board Layout . . . . . L 289
Plotting . . . . . L 291, U 92
Saving with SAVE . . . . . S 200–202
Schematic . . . . . L 224
Setting the bus address . . . . . I 19
Toggling with TMENU . . . . . S 221
Tag:
Field . . . . . U 97
Value . . . . . U 21, U 143–144, U 159, U Appendix
*TAN1 . . . . . L 28, L 30
*TAN2 . . . . . L 28, L 30, L 40, L 89
Temporary macros . . . . . U 66
Terminating a primary command . . . . . S 5
Terminating commands . . . . . L 25

Text editor:
Aborting commands . . . . . **M** 31
Adding text . . . . . **M** 29, **M** 102
Adjusting margins . . . . . **M** 45, **M** 93
Confirming commands . . . . . **M** 31
Copying text . . . . . **M** 40, **M** 95
Deleting text . . . . . **M** 34
Filling paragraphs . . . . . **M** 46, **M** 106
Formatting text . . . . . **M** 43, **M** 115
Moving text . . . . . **M** 40, **M** 95
Moving the cursor . . . . . **M** 32
Printing a file . . . . . **M** 47
Refreshing screen . . . . . **M** 31, **M** 120
Replacing character strings . . . . . **M** 37, **M** 111
Searching for patterns . . . . . **M** 42, **M** 99
Typing over characters . . . . . **M** 36, **M** 121
Using . . . . . **M** 27
Using to define macros . . . . . **S** 86
Text:
ADD . . . . . **S** 52–56
Changing slant with MODIFY . . . . . **S** 173
Changing with MODIFY . . . . . **S** 173
Loading stroke table with LOAD TSTROKE . . . . . **S** 160
Setting font size with FSIZE . . . . . **S** 126
Setting slant with TSLANT . . . . . **S** 224
Time:
Entering . . . . . **L** 8
Setting system time . . . . . **M** 85
Title block, updating . . . . . **L** 129, **L** 153
TMENU . . . . . **S** 221, **U** 89, **U** 94
TNAME . . . . . **S** 222
TNAME (archive) . . . . . **S** 296
Tolerance values . . . . . **S** 18–22
Tooling marks, adding . . . . . **L** 264
*Top30 . . . . . **L** 82
Traces:
Adding . . . . . **L** 263
Connecting . . . . . **L** 259, **L** 278

Tracing or flashing . . . . . U 181
Transferring files:
From magnetic/paper tape . . . . . M 19
To magnetic/paper tape . . . . . M 19
Transformations with PHOTOPLOT . . . . . U 186
Translate . . . . . M 61, M 157
TSLANT . . . . . S 224
TSLANT (archive) . . . . . S 297
Type-ahead buffer . . . . . U 46
Typing over characters in text editor . . . . . M 36, M 121

u

UNDO* . . . . . L 23
Unfilling an item . . . . . L 50
Ungathering a collection . . . . . S 127–130
Unit directory . . . . . M 159
Unit number (disc drive number) . . . . . I 5
Units . . . . . M 5
UNITS . . . . . S 226, U 12–13, U 53
UNITS (archive) . . . . . S 298–299
Unused space on a disc . . . . . S 67
Unwrapping instances with SMASH . . . . . S 207
Updating a title block . . . . . L 129, L 153
User restart . . . . . M 84
User:
Grid . . . . . L 18, U 14–15
Units . . . . . U 12–15, U 53
Written applications . . . . . U 46

v

Variables, calculating with EVALUATE . . . . . S 120
Verify . . . . . M 120
Verifying the part . . . . . L 191, L 247
Version . . . . . M 85
VERSION (Photoplot/Drill) . . . . . U 323
:VERTEX snapping mode . . . . . U 15
Vertex:
Adding . . . . . L 60
Adding with STRETCH . . . . . S 212–220
Snapping mode . . . . . S 4

Vertical dimension:
Adding . . . . . **L** 126
Line . . . . . **S** 19
Vertices, adjusting with a GATHER collection . . . . . **S** 127–132
VIA identifier . . . . . **U** 141, **U** 172
VIAs . . . . . **L** 281
Video interface card . . . . . **I** 21–22
Video output with DISPLAY . . . . . **S** 104–105
View, copying . . . . . **L** 133
VIEW (Photoplot/Drill) . . . . . **U** 323
Viewing instances . . . . . **L** 184
Viewport:
Description . . . . . **U** 4–6, **U** 59
Explanation . . . . . **L** 12, **L** 71
Specifying number with WINDOW . . . . . **S** 232
Specifying with SHOW . . . . . **S** 206
VIRTUAL DRILL (Photoplot/Drill) . . . . . **U** 325
VIRTUAL PHOTOPLOT (Photoplot/Drill) . . . . . **U** 325
VLCTY . . . . . **L** 38
Volume search table . . . . . **M** 9
Volume:
In search table . . . . . **S** 204–205, **U** 50–51
Name . . . . . **L** 37
Packing with PACK . . . . . **S** 179
Prefix . . . . . **L** 9
Specifying default volume with PREFIX . . . . . **S** 192
VOLUMES . . . . . **S** 228
Volumes:
Changing the default volume . . . . . **M** 54, **M** 150
Concepts . . . . . **M** 3
Listing . . . . . **M** 53, **M** 161
Names . . . . . **M** 4
Removing files . . . . . **M** 61, **M** 153
Storage capacity . . . . . **M** 7
$VR . . . . . **L** 17
*VRTX . . . . . **L** 261

W

Washer and nut, adding . . . . . **L** 136
*WDTH . . . . . **L** 26, **L** 262
What . . . . . **M** 87, **M** 162
WHEEL (Photoplot/Drill) . . . . . **U** 325
Wheel File:
  Aperture entries . . . . . **U** 308
  Aperture height . . . . . **U** 309
  Aperture name . . . . . **U** 308
  Aperture number . . . . . **U** 308
  Aperture shape . . . . . **U** 308
  Aperture type . . . . . **U** 309
  Aperture width . . . . . **U** 309
Width:
  Changing . . . . . **L** 51
  Changing with MODIFY . . . . . **S** 173
WIND* PAN . . . . . **L** 54
WIND* REDRW . . . . . **L** 31
WINDOW . . . . . **S** 230–233, **U** 5, **U** 19, **U** 78, **U** 92
Workfiles:
  Clearing . . . . . **M** 66, **M** 149
  Copying a file into . . . . . **M** 63, **M** 141
  Copying to a file from . . . . . **M** 63, **M** 141
  Displaying name . . . . . **M** 66, **M** 162
  Displaying state . . . . . **M** 66, **M** 162
  Executing . . . . . **M** 80
WRAP . . . . . **S** 234–237
WRAP* . . . . . **L** 64

X

Xchng . . . . . **M** 36, **M** 121
X,Y location errors, correcting . . . . . **L** 26

Z

Zap . . . . . **M** 35, **M** 123
Zero . . . . . **M** 163
Zero-base datum dimension . . . . . **S** 19
ZONE_ON . . . . . **L** 103
Zoom factor, setting with WINDOW . . . . . **S** 231
Zooming in on an item . . . . . **L** 35

HP Part Number
98305-90008
Microfiche No. 98305-99008
Printed in U.S.A. 11/86

For Internal Use Only